Universidade Federal do Espírito Santo

Centro Tecnológico

Programa de Pós-Graduação em Engenharia Elétrica

Marcus Vinícius Fitz Lucchetti

Identificação de pedestres por meio de mapas de densidade construídos com ASIFT e fluxo óptico

Vitória

2016

Marcus Vinícius Fitz Lucchetti

Identificação de pedestres por meio de mapas de densidade construídos com ASIFT e fluxo óptico

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.

Universidade Federal do Espírito Santo

Centro Tecnológico

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Patrick Marques Ciarelli

Vitória

2016

Marcus Vinícius Fitz Lucchetti

Identificação de pedestres por meio de mapas de densidade construídos com ASIFT e fluxo óptico

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica.

Trabalho aprovado. Vitória, 9 de dezembro de 2016:

Prof. Dr. Patrick Marques Ciarelli
Orientador

Prof. Dra. Raquel Frizera Vassallo - UFES

Prof. Dr. Jorge Leonid Aching Samatelo - UFES

Vitória
2016

Agradecimentos

Aos meus pais Irani Aparecida Fitz Lucchetti e Adilson Lucchetti que sempre me mostraram a importância do estudo contínuo, me suportando nos momentos de dificuldade.

Ao meu irmão Renan Emannuel Fitz Lucchetti, que mesmo distante fisicamente sempre me apoiou e me fez sentir uma pessoa especial em sua vida.

À minha companheira Renata Camatta Sodré, e ao meu enteado Luiz Eduardo Camatta Sodré Conde, que tiveram toda paciência e preocupação em me liberar de afazeres comum, me dando o tempo necessário para dedicação a este trabalho.

Ao Professor Patrick Marques Ciarelli, que dedicou muitas horas de seu tempo compartilhando conhecimento e técnicas, revisando este trabalho e artigos publicados, participando e apresentando esse tema em congressos, mas principalmente por acreditar em mim e me dar a oportunidade de desenvolver este projeto em parceria como orientado.

Às equipes responsáveis e revisores dos congressos CBA e WVC no ano de 2016, pela aceitação do tema deste trabalho e disponibilização de tempo para apresentações em sua agenda.

Aos meus gestores da empresa Vale que compreenderam em muitos momentos minha ausência na empresa para poder ir até às dependências da UFES participar das aulas que aconteceram durante o horário de trabalho.

À todos os professores, secretária e coordenador do PPGEE da UFES que de uma maneira ou outra compartilharam seu tempo comigo, e dedicaram seu conhecimento em prol do meu desenvolvimento.

Aos autores referenciados nas bibliografias deste trabalho que disponibilizaram seu conhecimento e ferramentas desenvolvidas que ajudaram a desenvolver a metodologia aqui exposta.

Muitas coisas não ousamos empreender
por parecerem difíceis; entretanto, são
difíceis porque não ousamos empreendê-las.
(Sêneca)

Resumo

Detecção de pessoas em imagens digitais é um paradigma de interesse pelas oportunidades de aplicação em temas como segurança e rastreamento de entidades. Recentes acontecimentos no mundo envolvendo a investigação de ações suspeitas em locais públicos como aeroportos, estações de metrô e manifestações com grande aglomerado de pessoas justificam o direcionamento de pesquisas no tema. Os diferentes estilos de roupas, cores, iluminação, articulações, oclusões intra-entre pessoas e a similaridade do contorno de objetos são alguns dos obstáculos a serem vencidos nos estudos. Neste sentido, este trabalho contribui com uma proposta para detecção de pessoas em grandes aglomerados de indivíduos. O método apresentado combina técnica de detecção de pedestres com mapas de densidades da concentração de possíveis indivíduos, em uma imagem digital, obtidos a partir de relevantes *keypoints* detectados com ASIFT e validados com fluxo óptico. Os *keypoints* validados permitem ainda uma etapa adicional de predição de pedestres, a qual é realizada *frame* a *frame* a partir da clusterização binária entre a possível localização de um pedestre e a saída do detector. O trabalho ainda propõe a generalização dos limiares de cálculo do mapa de densidade para permitir a integração da técnica de mapa de densidades com detectores de pedestres que retornam algum tipo de *score* de confiança da detecção. Os resultados dos experimentos indicam que essa abordagem é mais precisa que os métodos tradicionais.

Palavras-chave: Fluxo Óptico, Mapa de Densidade, Detecção de Pedestres, Processamento de Imagens, ASIFT, Clusterização Hierárquica Binária.

Abstract

Pedestrian detection in digital images is a relevant paradigm in areas such as security and entities tracking. Recent worldwide events involving investigation of suspicious actions in public places, such as airports, subway stations and crowd manifestations justify the direction of researches on this subject. Different clothing styles, colors, brightness, articulations, intra-inter people occlusions and similarities among boundaries of objects are some of the obstacles to be overcome by researchers. Under this perspective, this work contributes with a proposal to detect people in crowd. The presented method combines a people detection technique with a density map generated by relevant keypoints detected using ASIFT and validated with optical flow. The selected keypoints still allow an additional step of pedestrian prediction, which is performed frame by frame from binary clustering among the possible location of a pedestrian and the detector output. This research still purposes to generalize the density map thresholds in order to make the integration with pedestrian detectors that return some type of score. The experimental results indicate that this approach is more accurate than traditional methods.

Keywords: Optical Flow, Density Map, Pedestrian Detection, Imaging Processing, ASIFT, Binary Hierarchical Clustering.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

CCD	Dispositivo de Carga Acoplada (*Charge-Coupled Device*)
CMOS	Semicondutor de Metal-Óxido Complementar (*Complementary Metal Oxide Semiconductor*)
NMAE	Erro Absoluto Médio Normalizado (*Normalized Mean Absolute Error*)
MODP	Precisão da Detecção de Múltiplos Objetos (*Multiple Object Detection Precision*)
MODA	Assertividade da Detecção de Múltiplos Objetos (*Multiple Object Detection Accuracy*)
ASIFT	Transformada Afim das Características Invariantes à Escala (*Affine Scale Invariant Feature Transform*)
CBRLOF	*Cross-Based Robust Local Optical Flow*
MDPM	*Mixture Deformable Part Models*
HOG	Histogramas de Gradientes Orientados (*Histogram of Oriented Gradients*)
SVM	Máquina de Vetor de Suporte (*Support Vector Machine*)
DPM	Modelos de Partes Deformáveis (*Deformable Part Models*)
ConvNet	Rede Neural Convolucional (*Convolutinal Neural Network*)
L-SVM	*Latent Support Vector Machines*
LDCF	*Local Decorrelation Channel Features*
ACF	*Aggregated Channel Features*
CCCP	*Concave-Convex Procedure*
SIFT	Transformada de Características Invariantes à Escala (*Scale Invariance Feature Transform*)
DoG	*Difference of Gaussians*
ORSA	*Optimized Random Sampling Algorithm*

Sumário

1 Introdução

1.1 Motivação

As grandes inovações tecnológicas das últimas décadas foram alavancadas majoritariamente por um mesmo príncipio: a conversão das variáveis analógicas, captadas por sensores, em dados digitais. Uma das tecnologias que se beneficiou dessa evolução foram as câmeras digitais, as quais tiveram avanços a partir dos sensores de imagens *Charge Coupled Device* (CCD) e *Complementary Metal Oxide Semiconductor* (CMOS), que convertem luz em sinais elétricos analógicos, que por sua vez são convertidos em sinais digitais. A digitalização das imagens capturadas permitem a realização das pesquisas na área de processamento digital de imagens, a qual estabelece técnicas de manipulação em imagens para melhorar a qualidade visual para a interpretação humana, ou processamento da imagem para fins de armazenamento, transmissão e representação para interpretação por sistemas autônomos (GONZALEZ; WOODS, 2006). Dessa forma, em paralelo à evolução da conversão das imagens em bits, as soluções em *hardware* com alta capacidade de processamento proporcionaram assim o advento da visão computacional, a qual destina a computadores a tarefa de interpretar cenas a partir das matrizes de *pixels* capturadas, ditas *frames*, sendo a análise de vídeos uma das áreas de maior destaque por viabilizar trabalhos que auxiliam no dia a dia das pessoas e empresas.

No período de um dia, no mundo todo, uma grande quantidade de vídeos é gerada e disponibilizada em sistemas como aplicativos, redes sociais e monitoramento. Esse grande volume de dados impossibilita que seres humanos processem toda informação desejada, sendo que aplicações como detecção, rastreamento e segmentação de objetos, reconhecimento de padrões e faces, detecção de anomalias, carros autônomos, detecção de movimentações suspeitas ganharam sua importância no mercado.

A detecção e o rastreamento de objetos são tarefas fundamentais da visão computacional, sendo suas aplicações difundidas em diferentes oportunidades como sistemas de vigilância, controle de tráfego de veículos e diagnósticos médicos. Na linha de estudos direcionados à detecção de objetos em imagens, questões de segurança pessoal e patrimonial vem atraindo a atenção principalmente pela necessidade do monitoramento automatizado de pessoas em locais públicos. Eventos ao redor do mundo envolvendo a investigação de ações suspeitas em aeroportos e estações de trem ou ainda a estimativa de participantes em manifestações políticas, indicação de ameaças de tumultos, protestos violentos, brigas, pânico e agitação de massa são alguns dos exemplos mais recentes das possíveis aplicações (JUNIOR; MUSSE; JUNG, 2010).

Técnicas de detecção de pessoas são, em muitos casos, desenvolvidas para captura de um número reduzido de indivíduos em uma imagem. Quando tais técnicas são aplicadas em vídeos que contêm aglomerados de pessoas, acabam por gerar falsos positivos ou perdas em detecções por conta das alterações existentes nas silhuetas das pessoas por oclusões ou semelhança com outros objetos. Técnicas de cálculo de mapas de densidade de pessoas permitem avaliar as regiões das imagens onde existem maior probabilidade de existência de pessoas.

Neste trabalho, é esperado que a integração dessas técnicas, detecção de pessoas e mapa de densidade, em um ambiente com aglomeração de pessoas possa auxiliar na redução dos falsos positivos e, consequentemente, melhorar a acurácia final da saída do detector.

1.2 Objetivo

Busca-se neste trabalho desenvolver uma metodologia para detecção de pessoas em sequências de imagens que apresentem aglomeração de pessoas. Neste sentido, será avaliada a junção de técnicas de detecção de pessoas a mapas de densidade contruídos a cada sequencia de *frames* em vídeos. O vídeos utilizados para as técnicas propostas devem necesssariamenter ser capturados por câmera estática, ou seja, as imagens possuem *background* estático para o correto cálculo dos mapas de densidade. No método proposto, as saídas do detector de pedestre são validadas e filtradas pelo mapa de densidade na expectativa de aprimorar a precisão de saída do detector pela redução de falsos positivos.

No que tange a construção dos mapas de densidade, estes são calculados a partir da extração das características de interesse pela transformada afim de características invariantes à escala (ASIFT) (YU; MOREL, 2011), as quais são validadas por filtros que envolvem a presença da técnica *Cross Based Robust Local Optical Flow* (CBRLOF) (SENST et al., 2014).

Neste trabalho, busca-se estabelecer um método genérico de integração do mapa de densidade com diferentes tipos de detectores de pedestres, como aqueles usados em (FRADI; DUGELAY, 2015), (NAM; DOLLAR; HAN, 2014) e (FELZENSZWALB; MCALLESTER; RAMANAN, 2008).

Avalia-se, ainda, a contribuição de etapa adicional relacionada à "predição" de pessoas nas imagens em um dado *frame* I^k a partir das detecções do *frame* I^{k-1} . Tais predições são geradas pela contribuição das próprias características locais extraídas das imagens e do fluxo óptico entre esses *frames*, sendo utilizada a clusterização hierárquica binária para validação e remoção de detecções duplicadas.

Ao final, uma proposta de integração entre os diferentes detectores é realizada,

avaliando qual a contribuição existente quando dois detectores distintos são utilizados para reforçar os resultados de detecção final de pedestres.

1.3 Caracterização do Problema

A detecção de pessoas tem sido extensivamente estudada nas últimas décadas. Embora um grande número de estudos tenham sido propostos, ainda existem problemas que prejudicam a viabilidade comercial de muitas das soluções.

Dentro do que pode ser encontrado na literatura, e nas bases de dados públicas disponíveis para a identificação de pessoas em imagens digitais, as pessoas aparecem em diferentes posições e formas, devido à quantidade de pessoas em uma imagem, ângulo de visão, iluminação, diversidade de roupas ou *background* (RODRIGUEZ et al., 2011). Grande parte das bases de dados e dos trabalhos já desenvolvidos utilizam-se da localização de pessoas isoladas dentro do contexto de uma imagem digital, procurando aprimorar os indicadores de detecção de pedestres em diferentes perspectivas e escalas. Em um outro segmento de trabalho, os estudos são realizados em imagens onde há alta concentração de pessoas. Esta segunda abordagem apresenta novos desafios, os quais relacionam-se às inter-oclusões ou auto-oclusões existentes nas partes do corpo das pessoas, ou ainda nas falsas detecções onde confunde-se a detecção de uma pessoa com um ombro de outra, ou ainda assemelha-se a cabeça ou parte do corpo de um indivíduo a um outro objeto (ZHANG; BAUCKHAGE; CREMERS, 2014). As imagens apresentadas na Figura 1 ilustram algumas situações encontradas no reconhecimento de pessoas em imagens com alta densidade de pedestres.

Ainda sobre os desafios catalogados na identificação de pedestres, verifica-se que para cada técnica utilizada como detector de pessoas, as dificuldades se diferem. Exemplo disso são os métodos que utilizam cores, tamanhos, formas, posições ou movimentos para detecção, com seus desempenhos relacionados aos desafios da característica de sua concepção (FU et al., 2015).

1.4 Bibliografia Relacionada

1.4.1 Detecção de Pessoas

Existe uma grande quantidade de trabalhos na literatura e técnicas desenvolvidas sobre o tema de detecção de pessoas em imagens digitais. O clássico descritor HOG (*Histogram of Oriented Gradients*), descrito em maiores detalhes na Seção 2.1, é um dos descritores mais difundidos, e se baseia em diferenças locais da imagem sendo proposto em (DALAL; TRIGGS, 2005). O descritor HOG ainda ficou conhecido por ser base para outros trabalhos que o utilizam para o densenvolvimento de novas abordagens, como

(a) sobreposição de pessoas (b) alta concentração de pessoas

(c) pessoas em posição diferentes (d) pessoas com ”corte” de membros

Figura 1 – Exemplo do posicionamento de pessoas em diferentes cenas.

na técnica de partes móveis deformáveis, Seção 2.3, proposta em (FELZENSZWALB; MCALLESTER; RAMANAN, 2008) , ou no seu aperfeiçoamento, como em (ZHANG et al., 2015) e (NAM; HAN; HAN, 2011).

Existem ainda outras variações do HOG que, por exemplo, não utilizam a etapa de normalização, como em (DOLLAR et al., 2009) e (BENENSON et al., 2013). Algumas técnicas ainda são conhecidas como uma extensão do HOG, como a técnica *Local Binary Pattern* desenvolvidas em (WANG; HAN; YAN, 2009), (WANG et al., 2015), onde mesmo trabalhando com as magnitudes dos *pixels*, a diferença está na maneira como esta técnica trata a informação do gradiente, fazendo uso de 8 (oito) direções para cada *pixel*, e não apenas uma direção como apresentado em (DALAL; TRIGGS, 2005).

Outras abordagens ainda herdaram os princípios de HOG e têm suas contribuições variadas utilizando características diversas das imagens, como: 1) cores, como pode ser visto em (KHAN et al., 2012) e (KHAN et al., 2013), 2) estruturas locais, que ao invés de *pixels* são utilizadas entidades maiores por similaridade, como tratado em (WOLF; HASSNER; TAIGMAN, 2010), 3) segmentação, (OTT; EVERINGHAM, 2009) e 4) estimativa de

fronteiras locais (LIM; ZITNICK; DOLLÁR, 2013). Além de outras técnicas que se utilizam da covariância entre características de cor, gradiente e orientação (PAISITKRIANGKRAI; SHEN; HENGEL, 2014) e (TUZEL; PORIKLI; MEER, 2008).

Trabalhos recentes relacionados à detecção de pessoas também incluem a concepção de partes móveis deformáveis e extensões dessa técnica, como em (PARK; RAMANAN; FOWLKES, 2010) e (YAN et al., 2013), redes neurais convolucionais e *deep learning*, (SERMANET et al., 2012) e (ZENG; OUYANG; WANG, 2013) e abordagens que focam na otimização e aprendizado (LEVI; SILBERSTEIN; BAR-HILLEL, 2013). Detectores impulsionados também são amplamente utilizados, em particular, os detectores de características de canais como proposto em (BENENSON et al., 2013), (DOLLAR et al., 2014), (BENENSON et al., 2012) e (DOLLAR et al., 2009), os quais são uma família de detectores baseados em árvores de decisão impulsionadas, e calculadas sobre múltiplos canais de características, tais como cor, gradiente de magnitude, gradiente de orientação entre outros.

1.4.2 Mapa de Densidade

Na última década, de uma forma geral, para estimar a densidade de pessoas em imagens digitais, foi empregada a realização de uma etapa de extração de características da imagem que representem de maneira coerente a informação desejada, no caso do escopo deste trabalho: a informação de pessoas. Em seguida, visando encontrar a densidade de pessoas em um ambiente, a partir das características extraídas, foram utilizadas técnicas como treinamento de classificadores, fluxo óptico e redes neurais. Resumidamente, os esforços se concentram na extração das melhores características das imagens e nos testes de melhores técnicas para realização da tarefa de estimar a densidade de pessoas.

Em (DAVIES; YIN; VELASTIN, 1995) é proposto tanto o cálculo da densidade de pessoas estacionárias, pela remoção do *background* e detecção de bordas (*edge-detection*), quanto a estimativa de movimento de pessoas por meio de fluxo óptico. Essa técnica apresentou-se pouco efetiva em situações onde a densidade de pessoas é muito grande.

Para lidar com uma densidade de pessoas maior, (MARANA et al., 1997) trouxeram a utilização da informação de textura da imagem baseada na matriz de dependência de níveis de cinza (GLDM - *Gray Level Dependency Matrix*), sendo que as características extraídas por esse método foram utilizadas em uma rede neural auto-organizada, que gera como resposta um mapa de densidade de pessoas. Já em (XIAOHUA; LANSUN; HUANQIN, 2006), (MA; HUANG; LIU, 2010) e (ZHOU; ZHANG; PENG, 2012) a técnica de classificação SVM (Seção 2.2.1) foi utilizada com o objetivo de melhorar o desempenho dos cálculo do mapa de densidade.

Em (ALI; DAILEY, 2012) é combinado o detector Viola e Jones (VIOLA; JONES, 2001) com a eliminação de falsos positivos pelo método de mínimos quadrados não linear,

e um classificador Adaboost cascateado para o rastreamento de pessoas em ambientes de grandes oclusões a partir do mapeamento de cabeças de indivíduos detectados nas imagens.

Na tendência de trabalhos com extração de características locais, em (YANG et al., 2012) é utilizado o código SST-LBP (*Sparse Spatio-temporal Local Binary Parttern*) como extrator. Este extrator separa as informações em duas categorias, temporais e espaciais, que são relacionadas ao final do processo por meio de um SVM (*Support Vector Machine*), cuja saída apresenta o nível de densidade de pessoas por região da imagem.

Soluções que utilizam fluxo óptico na determinação do mapa de densidade de pessoas em *frames* sequenciais, como em (RAO et al., 2014) e (FRADI; DUGELAY, 2015), não necessitam de treinamento prévio para detecção de características distintas e aplicação de técnicas para eliminação de ruídos inerentes à captura das imagens. Abordagens recentes ainda justificam a utilização de redes neurais convolucionais, como em (FU et al., 2015), que traz a possibilidade de aprendizado de características pela alteração dos pesos das conexões da rede.

1.5 Método Proposto

A metodologia proposta baseia-se principalmente pela integração existente entre a saída dos detectores de pedestres. Esta integração acontece pelas filtragens realizadas a partir de mapas de densidade construídos para oferecer uma maior precisão ao resultado final das detecções pela redução de falsos positivos. No entanto, existem modificações adicionais propostas e implementadas entregues como produto deste trabalho, além do que já existe na literatura sobre este processo de integração, as modificações são:

1. Construção de Mapas de Densidade a partir da extração de características pela técnica ASIFT.
2. Normalização dos *scores* de detecção de detectores de pedestres distintos.
3. Cálculo de limiares máximos e mínimos adaptativos aos *scores* para construção dos mapas de densidade.
4. Predição de hipóteses de detecção de pessoas para o *frame* I^k, a partir das hipóteses do *frame* I^{k-1} e das informações de Fluxo Óptico.
5. Integração entre as saídas dos detectores após as filtragens realizadas.

A Figura 2 resume os principais processos implementados neste trabalho para a detecção de pessoas e indica a localização de cada modificação proposta enumeradas acima.

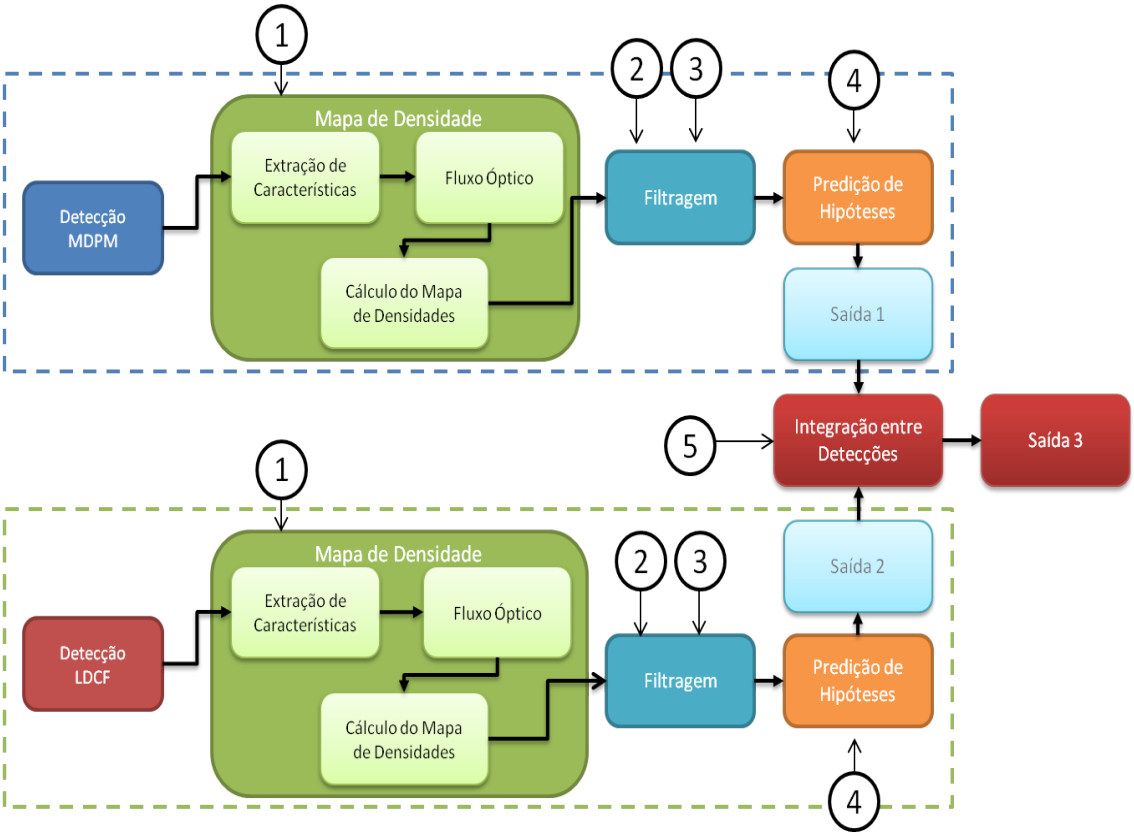


Figura 2 – Diagrama indicando a localização das modificações propostas em cada etapa do processo.

1.6 Artigos Publicados

LUCCHETTI, M. V. F.; CIARELLI, P. M. Combining Density Map and Predict Detections for Pedestrian Detection in Crowds In: WORKSHOP DE VISÃO COMPUTACIONAL, 2016, Campo Grande.

LUCCHETTI, M. V. F.; CIARELLI, P. M. Identificação de pedestres por meio de mapas de densidade construídos com ASIFT e fluxo óptico In: Congresso Brasileiro de Automática: CBA, 2016, Vitória.

1.7 Estrutura da Dissertação

A estrutura do trabalho até aqui apresentou de forma geral o tema proposto, suas motivações e escopo. O restante do trabalho está dividido de forma a detalhar a abordagem desenvolvida.

O Capítulo 2 aborda definições sobre a detecção de pedestres e explora em maiores detalhes o clássico extrator de características HOG; uma das principais técnicas de classificação encontradas na literatura, o SVM (*Support Vector Machine*), e uma derivação

dessa técnica conhecida como L-SVM (*Latent Support Vector Machine*). Ainda neste capítulo é descrito os dois detectores de pedestres utilizados para os experimentos deste trabalho: 1) Mistura de Modelos de Partes Deformáveis (MDPM) e 2) Descorrelação Local de Canais de Características (LDCF).

Na sequência, o Capítulo 3 traz as principais definições esperadas para o cálculo do mapa de densidade que será integrado à saída dos detectores de pedestres. Dessa forma, são apresentados os principais extratores de características utilizados neste trabalho, além da técnica de fluxo óptico empregada e como a união dessas técnicas viabilizam a construção de mapas de densidade a partir de dois *frames* consecutivos de um vídeo.

O Capítulo 4 descreve a metodologia proposta para a integração e filtragem dos resultados de saída dos detectores de pedestres e os mapas de densidades calculados. Este capítulo ainda descreve passos adicionais experimentados que visam aprimorar a maior robustez dos resultados e fornecer adaptabilidade de outros detectores à mesma técnica de integração pelo cálculo dinâmico de limiares para a construção dos mapas de densidade.

Os experimentos realizados no decorrer deste trabalho e os resultados relacionados são apresentados e comentados no Capítulo 5, onde é possível observar as contribuições dos métodos propostos neste trabalho.

Finalmente, o Capítulo 6 é constituído das conclusões identificadas a partir dos experimentos e resultados realizados, e ainda evoluções sugeridas para estudos futuros.

2 Detecção de Pedestres

A detecção de pedestres é um dos pilares mais explorados quando falamos em detecção de objetos em visão computacional. Devido à sua possibilidade de uso em aplicações como segurança automobilística, sistemas de vigilância e robôs, o tema tem atraído bastante a atenção de pesquisadores nos últimos anos, sendo que as pesquisas disponíveis empregam uma série de abordagens para o tratamento de detecção de pessoas.

A constante evolução sobre cada nova técnica publicada é tão importante quanto o emprego de técnicas inovadoras, ou seja, evoluir o que já existe é tão importante quando criar técnicas ainda não existentes. Exemplos dessas alterações são: base de dados de treinamento, métodos de classificação, mistura de técnicas, extratores de características, entre outros (BENENSON et al., 2015).

Um breve histórico da evolução das técnicas de detecção parte desde as variações da técnica proposta por (VIOLA; JONES, 2004), com a seleção de características *Haar* e criação das imagens integrais; passando pela utilização de descritores de características extraídas das imagens proposta por (DALAL; TRIGGS, 2005) com a técnica de Histogramas Orientados a Gradientes (do inglês, *Histogram of Oriented Gradients* - HOG), em conjunto com uma Máquina de Vetor de Suporte (do inglês, *Support Vector Machine* - SVM); evoluindo para os detectores baseados em Modelos de Partes Deformáveis (do inglês, *Deformable Part Models* - DPM) publicado por (FELZENSZWALB; MCALLESTER; RAMANAN, 2008); até os atualmente difundidos detectores por redes neurais convolucionais (CNN) visto em (SERMANET et al., 2012).

Neste trabalho são utilizadas duas técnicas de detecção de pedestres para auxiliar no reconhecimento de indivíduos em grupo de pessoas: 1) o detector genérico proposto por (FORSYTH, 2014), conhecido como Mistura de Modelos baseados em Partes Deformáveis (MDPM), o qual pode ser treinado para classificar hipóteses de pessoas e 2) a técnica de Detecção de Pedestres por Descorrelação Local de Canais de Características (LDCF) 2.4 proposta em (NAM; DOLLAR; HAN, 2014), que utiliza árvores de decisão e que parte de uma técnica antecessora conhecida como ACF (*Aggregated Channel Features*)que foi proposta em (DOLLAR et al., 2014).

As seções seguintes apresentam técnicas clássicas utilizadas na detecção de pedestres em imagens digitais, iniciando-se pela apresentação da técnica de descritores HOG, passando pelos classificadores SVM com seus derivados, até a descrição dos detectores de pedestres usados, que são apresentados em maiores detalhes nas Seções 2.3 e 2.4.

2.1 Descritores HOG

A técnica de extração de característica HOG (Gradientes Orientados a Histograma, do inglês, *Histrogram of Oriented Gradients*) descreve a aparência e formato de um objeto pela distribuição dos gradientes de intensidade. Uma imagem é dividida em pequenas porções conectadas chamadas "células" que contém um conjunto de *pixels*. Para cada célula então calcula-se um histograma de gradientes. Os descritores são representados no final pela concatenação desses histogramas de gradientes encontrados para cada "célula". A fim de obter uma melhor resposta à variação de iluminação e sombras, os histogramas são normalizados em termos de contrastes pelo cálculo da intensidade de regiões maiores chamadas "blocos".

2.1.1 Gradientes de Intensidade

Considerando que o gradiente de intensidade de um *pixel* (x,y) é representado por uma magnitude $r(x, y)$ e uma orientação $\theta(x, y)$, o primeiro passo do HOG é encontrar os gradientes de intensidade da imagem aplicando as máscaras espaciais [-1 0 1] na direção horizontal e a máscara $[-1\ 0\ 1]^{\mathrm{T}}$ na direção vertical sobre a imagem. Quando se trabalha com imagens coloridas, calcula-se o gradiente de intensidades para cada canal de cor (RGB), e o gradiente de intensidade final $r(x, y)\angle\theta(x, y)$, para cada *pixel*, é o gradiente com a maior magnitude $r(x, y)$ entre os canais de cor.

2.1.2 Histograma de Gradientes

O HOG utiliza janelas de detecção de dimensões 64×128 *pixels*. Para os descritores, opera-se em células de 8×8 *pixels* pertencentes às janelas de detecção. Essas células são organizadas em blocos. A Figura 3 ilustra uma janela de detecção evidenciando uma célula de 8×8 *pixels* pelo quadrado vermelho.

Em cada célula, calcula-se o vetor gradiente a cada *pixel*. Todos os 64 vetores de gradientes encontrados são distribuídos em um histograma de graus $\times$ magnitude, com um resultado similar ao da Figura 4. Este histograma é conhecido com histograma $9 - bin$, pelas suas 9 divisões no eixo dos graus, ou seja, 20 graus por *bin*. Os vetores gradientes são calculados *pixel* a *pixel* e contêm informações de magnitude e direção (graus) considerando as intensidades dos *pixels* vizinhos na célula.

2.1.3 Normalização dos Histogramas

A partir deste momento, com o cálculo dos vetores gradientes, deve-se normalizar o histograma de gradientes. Esse processo ocorre para que os gradientes fiquem mais robustos à variações de contrastes, e ocorre pela divisão dos vetores pela suas respectivas magnitudes. Dividir um vetor pela sua magnitude é como dizer que se está normalizado

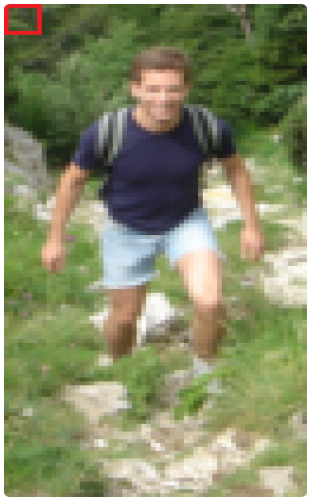


Figura 3 – Exemplo de uma janela de detecção 64 × 128.

este vetor ao tamanho unitário, pois o resultado do vetor terá magnitude 1. Normalizar um vetor não afeta sua orientação, apenas a magnitude. Da mesma forma, a normalização do histograma pode garantir a robustez quanto à iluminação, tornando a técnica mais robusta.

2.1.4 Normalização do Bloco

Embora cada histograma individual seja normalizado, as células são agrupadas em blocos e normalizadas baseadas em todos os histogramas do bloco.

O bloco comumente usado, como mostrado em (DALAL; TRIGGS, 2005), são formados de 2 × 2 células. E ainda, os blocos possuem uma sobreposição de 50%, assim como ilustrado na Figura 5.

A normalização do bloco é realizada pela concatenação dos histogramas, das 4

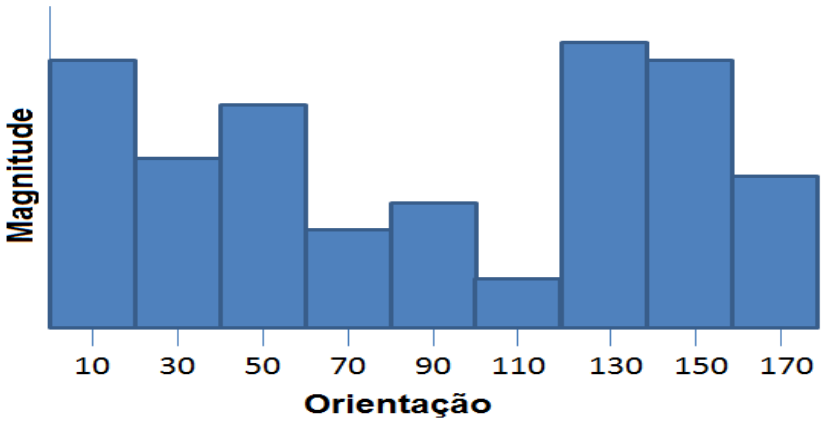


Figura 4 – Exemplo de Histograma 9 – *bin*.

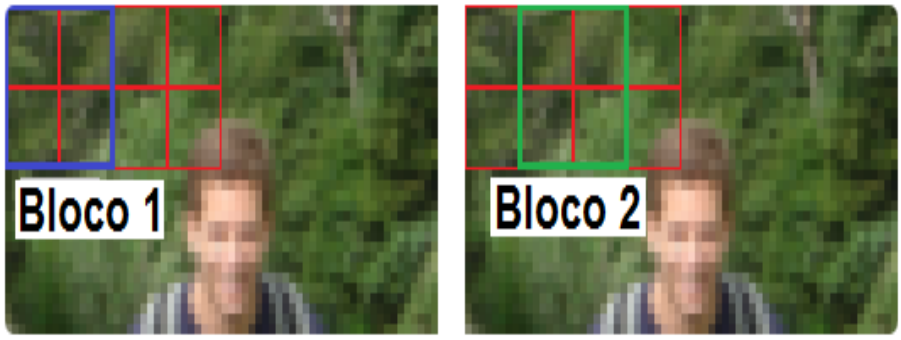


Figura 5 – Exemplo da sobreposição dos blocos em HOG.

células dentro do bloco, em 36 componentes (4 histogramas × 9 *bins* por histograma). A normalização ocorre pela divisão do vetor concatenado pela sua magnitude.

O efeito da sobreposição dos blocos infere que, ao final, um mesmo *pixel* irá aparecer múltiplas vezes no descritor final, mas normalizado sempre por um conjunto diferente de células vizinhas. Resumidamente, células de canto da imagem aparecerão uma vez, células das bordas aparecerão duas vezes, e as células do interior, quatro vezes.

Seja $\mathbf{v}$ o descritor não normalizado, $\|\mathbf{v}\|_k$ sua k-ésima norma, e ϵ uma pequena constante que evita divisões por zero. A normalização de melhor performance proposta por (DALAL; TRIGGS, 2005) é a normalização L2-Hys, a qual é seguida de limitação dos valores de $\mathbf{v}$ em 0,2 e renormalização L2.

2.1.5 Descritor Final

A janela de detecção 64 × 128 *pixels* é dividida em 7 blocos horizontais e 15 blocos verticais, um total de 105 blocos. Cada bloco contém 4 células com um histograma $9 - bin$ para cada célula, um total de 36 valores por bloco. Isso gera um vetor final de 7 blocos horizontais × 15 blocos verticais × 4 células por bloco × histogramas de $9 - bin = 3780$ valores.

2.2 Classificadores

Uma vez extraídos os descritores para representar uma imagem, é necessário usar uma técnica de aprendizado de máquina para informar se na região analisada existe um pedestre ou não. Técnicas de Aprendizado de Máquina estão relacionadas aos trabalhos que procuram desenvolver métodos com a capacidade de "adquirir conhecimento" a partir de exemplos prévios (MITCHELL, 1997). Em outras palavras, são técnicas que a partir de experiências passadas (base de dados) podem criar uma hipótese (conjunto de regras) ou função capaz de resolver um problema, utilizando-se de um princípio chamado de indução

de hipóteses.

Uma área do campo de Aprendizado de Máquina é focado em utilizar algoritmos como ferramentas de classificação para um conjunto de exemplos de entrada. A classificação é interpretada como sendo o processo de rotular uma determinada informação de entrada a uma classe, a qual ela supostamente faz parte (RUSSELL; NORVIG, 2003). Dessa forma, os algoritmos de Aprendizado de Máquina são utilizados no desenvolvimento de uma classificador que, a partir de um conjunto de treinamento, é capaz de direcionar as instâncias de entrada para uma das classes para o qual foi treinado.

Existem dois tipos básicos de aprendizado para que um algoritmo encontre uma hipótese, são eles: aprendizado *supervisionado* e aprendizado *não-supervisionado* (HAYKIN, 1998). A escolha do método de aprendizado interfere na maneira como a técnica irá se relacionar à base de exemplos para predizer a classificação. Neste trabalho são utilizadas técnicas de Aprendizado de Máquina com aprendizado supervisionado.

No aprendizado *supervisionado* busca-se encontrar uma função ou hipótese que aproxime o máximo possível um conjunto de dados exemplos (entrada) de um conjunto de valores desejados (saída). Aqui pode-se entender que no *aprendizado supervisionado* existe a figura de um “professor", que mostra dados conhecidos (dados rotulados) na forma de entrada-saída.

Já no aprendizado *não-supervisionado* não existe o conhecimento do conjunto de saída rotulados. Neste método faz-se necessário reconhecer padrões existentes nos dados para inferir os melhores agrupamento de dados.

2.2.1 SVM

Dentro do escopo de aprendizado *supervisionado* encontramos uma técnica baseada na Teoria de Aprendizado Estatístico (CORTES; VAPNIK, 1995) que ganhou destaque em trabalhos de aprendizado de máquinas, que são as SVM (Máquinas de Vetores de Suporte, do inglês, *Support Vector Machines*). Os resultados obtidos pelas SVMs possuem desempenhos satisfatórios em tarefas do dia a dia como a detecção de faces e pessoas em imagens digitais, reconhecimento de caracteres, aplicações na área de bioinformática, robótica e outros.

Vale ressaltar duas características encontradas nas respostas das SVMs (SMOLA; BARTLETT, 2000) que corroboram com sua utilização em trabalhos com imagens digitais:.

- **Boa Capacidade de Generalização:** as SVMs se mostram robustas quando na tarefa de classificação de dados, predizem bem novos dados que não estão necessariamente nas amostras de treino. Essa capacidade de generalização é dita

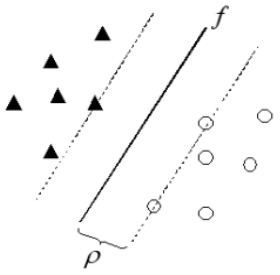


Figura 6 – Exemplo de um possível hiperplano $f_\beta(x)$ que separa um conjunto de dados.

robusta quanto a *overfitting* pois os exemplos não são "decorados", ou seja, o classificador não é especializado apenas no conjunto de treinamento.

- **Robustez em grandes dimensões:** as SVMs se comportam bem com dados de grandes dimensões como é o caso de imagens.

2.2.1.1 SVMs Lineares com Margens Rígidas

A proposta inicial das SVMs é o tratamento de classificações binárias, ou seja, que envolvem apenas duas classes. A mais simples dessas propostas é a classificação de conjuntos linearmente separáveis, ou SVMs com margens rígidas. Um conjunto linearmente separável é composto por exemplos que podem ser separados por pelo menos um hiperplano (RUSSELL; NORVIG, 2003). As SVMs lineares buscam o hiperplano ótimo, por minimização de algumas equações sob determinadas restrições, e que representam a margem de separação entre os dados a serem classificados de forma maximizada.

A margem de um classificador linear é definida como a menor distância entre os exemplos do conjunto de treinamento e o hiperplano encontrado na separação desses dados em classes.

Considere que cada ponto da Figura 6 representa uma imagem, e que cada formato de ponto representa uma classe distinta. Deseja-se separar o banco de imagens em duas classes. Procura-se então encontrar qual a linha de fronteira (hiperplano ($f_\beta(x)$) que melhor segrega os dois tipos de imagens, e sua respectiva margem ρ. Os pontos que se encontram sobre a linha de margem são conhecidos como Vetores de Suporte.

O hiperplano que separa esse conjunto linearmente separável pode ser definido pela Equação 2.1:

$$f_\beta(x) = \beta \cdot \mathbf{x} + b = 0, \tag{2.1}$$

onde $\beta \cdot \mathbf{x}$ é o produto escalar entre os vetores β e $\mathbf{x}$, sendo β o vetor normal ao hiperplano f, e b é um termo compensador. O par (β, b) é determinado durante o treinamento do

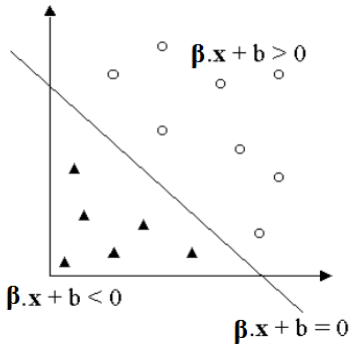


Figura 7 – Exemplo de hiperplano separador.

classificador. Dessa forma, a Equação 2.1 divide o conjunto de dados em duas classes: classe $y_i = +1$, se $\beta \cdot \mathbf{x} + b > 0$ e classe $y_i = -1$, se $\beta \cdot \mathbf{x} + b < 0$, como pode ser visto na Figura 7. Pode-se então definir a Equação 2.2 como uma função sinal $g(x) = sgn(f_\beta(x))$ que classifica corretamente todos os exemplos no conjunto S. Se $f_\beta(x) = 0$, $g(x)$ é desconhecido.

$$g(x) = sgn(f(x)) = \left\{ \begin{matrix} +1 & se & f_\beta(x) > 0 \\ -1 & se & f_\beta(x) < 0) \end{matrix} \right\}, \tag{2.2}$$

Um conjunto de treinamento S é linearmente separável se for possível determinar pelo menos um par (β, b) tal que a função sinal $g(x)$ consiga classificar corretamente todos os exemplos contidos neste conjunto S.

Seja f uma hipótese utilizada para classificação de entradas na forma (x_i,y_i), onde y_i representa a classe do padrão x_i. A margem ρ_f com o qual o padrão x_i é classificado é determinada pela Equação 2.3, sendo que a margem do classificador é definida pela Equação 2.4, obedecendo a necessidade de minimizar o risco empírico maximizando o valor da margem ρ. O hiperplano que possui esse valor de ρ maximizado é denominado *hiperplano ótimo*.

$$\rho_f(\mathbf{x_i}, y_i) = y_i(f_\beta(\mathbf{x}_i)) \tag{2.3}$$

$$\rho = min(y_i(f_\beta(\mathbf{x}_i)) \tag{2.4}$$

2.2.1.2 SVMs Lineares com Margens Suaves

A extensão das SVMs de margens rígidas para a classificação de conjuntos mais gerais pode ser realizada por meio de uma suavização das restrições impostas nas margens de separação entre os dados. Essa otimização das marges rígidas é importante pois, na

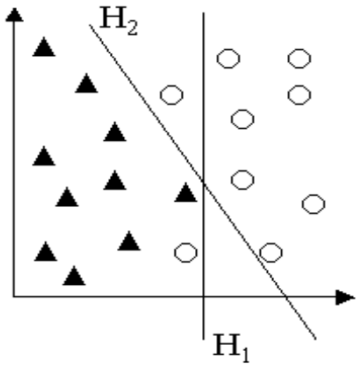


Figura 8 – Exemplo de conjunto não linearmente separável com dois hiperplanos H_1 e H_2 gerados sobre o conjunto.

prática, poucas são as aplicações onde os conjuntos são linearmente separáveis devido à presença de ruídos nos dados ou à própria natureza do problema, que pode ser até não-linear, sendo necessário a utilização de outras técnicas. A suavização das margens agora introduz um conceito de aceitação da ocorrência de alguns erros de classificação.

Nas SVMs lineares com margens rígidas é suposto que a determinação do par (β,b) é calculada, tal que a função sinal $g(x)$ tenha capacidade de classificar corretamente todos os exemplos contidos no conjunto S. Já no caso de conjuntos não linearmente separáveis, Figura 8, é mais difícil o atendimento a essa condição de classificação sem erros.

De maneira a tratar esses casos, aplica-se o conceito de variáveis de relaxamento ξ, definidas pelas Equações 2.5 e 2.6. Essa variáveis de relaxamento suavizam as restrições impostas na determinação do hiperplano ótimo, sendo admitidas ocorrências de erros de classificação.

$$\text{Para} \quad y_i = +1 \quad \xi_i(\beta, b) = \begin{Bmatrix} 0 & \text{se} \quad \beta \cdot \mathbf{x}_i + b \geqslant 1 \\ 1 - \beta \cdot \mathbf{x}_i + b & \text{se} \quad \beta \cdot \mathbf{x}_i + b < 1 \end{Bmatrix} \tag{2.5}$$

$$\text{Para} \quad y_i = -1 \quad \xi_i(\beta, b) = \begin{Bmatrix} 0 & \text{se} \quad \beta \cdot \mathbf{x_i} + b \leq -1 \\ 1 + \beta \cdot \mathbf{x}_i + b & \text{se} \quad \beta \cdot \mathbf{x}_i + b > -1 \end{Bmatrix} \tag{2.6}$$

As variáveis ξ_i medem onde se encontram os exemplos ($\mathbf{x_i}$,y_i) em relação aos hiperplanos $\beta \cdot \mathbf{x} + b \pm 1$. Se $\xi_i = 0$, o exemplo está fora da região entre os hiperplanos definidos por $\beta \cdot \mathbf{x} + b \pm 1$, então é classificado corretamente. Se $\xi_i > 0$, então temos o valor da distância entre o padrão $\mathbf{x_i}$ e o hiperplano $\beta \cdot \mathbf{x} + b$=0. Já se $\xi_\mathbf{i} > 1$ o dado é classificado erroneamente.

As variáveis de relaxamento ξ_i devem ter um valor mínimo para todo o conjunto de treinamento. A redução do risco empírico é dado pela minimização de $\|\beta\|$. Sendo assim

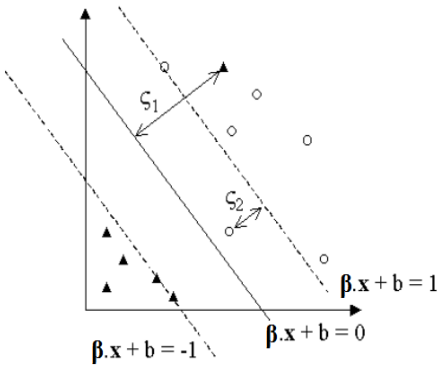


Figura 9 – Variáveis de relaxamento (ξ_i) aplicadas a um conjunto S.

os valores a serem minimizados são combinados na Equação 2.7.

$$f(\beta, b) = \beta^T \beta + C \sum_{i=1}^{n} \xi_i(\beta, b), \tag{2.7}$$

onde C é uma constante que impõe um peso diferente para o treinamento em relação à generalização e deve ser determinada empiricamente e n a quantidade de exemplos de treinamento.

2.2.1.3 Otimização sobre β e ξ

A determinação de $f(\beta, b)$ nos dará o Hiperplano Ótimo de margens suavizadas. A Equação 2.7 está submetida às seguintes restrições apresentadas na Equação 2.8.

$$\begin{cases} \xi_i \geqslant 0 \\ y_i(\beta \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, ..., n \end{cases} \tag{2.8}$$

Utilizando a Equação 2.1, a restrição $y_i(\beta \cdot \mathbf{x}_i + b) \geq 1 - \xi_i$ pode então ser reescrita como:

$$y_i f_\beta(\mathbf{x}_i) \geq 1 - \xi_i \tag{2.9}$$

Considenrado a restrição $\xi_i \geqslant 0$ apresentada na Equação 2.8, podemos reapresentar a Equação 2.9, função perda, da seguinte forma:

$$\xi_i = max(0, 1 - y_i f_\beta(\mathbf{x}_i)) \tag{2.10}$$

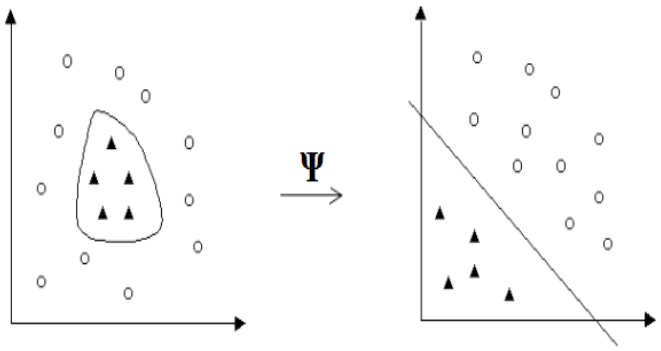


Figura 10 – Mapeamento do conjunto de dados de S para um novo domínio denominado *espaço de carcterísticas*, por meio da função Ψ_M.

Assim, o problema de aprendizagem resume-se à minimização da função objetivo L_S sobre β otimizada, sem restrição, apresentado na Equação 2.11.

$$L_S(\beta) = \frac{1}{2}\beta^T\beta + C\sum_{i=1}^{n} max(0, 1 - y_i f_\beta(\mathbf{x}_i)) \tag{2.11}$$

2.2.1.4 SVMs Não Lineares

Existem ainda casos onde não é possível dividir satisfatoriamente dados de treinamento por um único hiperplano, mesmo aceitando erros como visto nas SVMs lineares de margens suaves (Seção 2.2.1.2). Sendo assim é possível generalizar as SVMs lineares para que essas situações sejam tratadas.

Para que este problema de classificação seja contornado, cria-se agora o conceito de *espaço de características*. Para isso, definem-se funções reais $\Psi_1, ..., \Psi_M$ no domínio do espaço de entrada. Essas funções então são utilizadas para o mapeamento do conjunto de treinamento S para o novo espaço. A Figura 10 apresenta essa transformação.

A utilização do *espaço de características* a partir da escolha adequada de uma função Ψ, pode tornar o conjunto de treinamento linearmente separável. Assim, as SVMs apresentadas nas Subseções 2.2.1.1 e 2.2.1.2 podem então ser utilizadas sobre o conjunto de treinamento mapeado no novo espaço (HEARST, 1998).

As equações 2.12 e 2.13 apresentam as definições de S e Ψ, respectivamente. A dimensão M da função Ψ pode ser muito maior que m do espaço de $\mathbf{x}$.

$$\begin{gathered} S = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), ..., (\mathbf{x_n}, y_n)\} \\ \text{em que} \quad \mathbf{x_i} = (x_{i1}, x_{i2}, ..., x_{im}) \text{ e } y_i \epsilon \{-1, +1\} \end{gathered} \tag{2.12}$$

Tabela 1 – Kernels mais comuns utilizados nas SVMs (HAYKIN, 1998).

Tipo de *Kernel*	Função $K(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{\Psi}(\mathbf{x_i}) \cdot \mathbf{\Psi}(\mathbf{x_j})$
Polinomial	$(\mathbf{x_i}^T \cdot \mathbf{x_j} + 1)^p$
Gaussiano	$e^{-\frac{1}{2\sigma^2}\left\|\mathbf{x_i}-\mathbf{x_j}\right\|^2}$
Sigmoidal	$tahn(\eta_0 \mathbf{x_i} \cdot \mathbf{x_j} + \eta_1)$

$$\begin{aligned} &\mathbf{x_i}(i_1, ..., n) \mapsto \Psi(\mathbf{x_i}) = (\Psi_1(\mathbf{x_i}), \Psi_2(\mathbf{x_i}), ..., \Psi_M(\mathbf{x_i})) \\ &\Rightarrow \Psi(S) = \{\mathbf{\Psi}(\mathbf{x_1}), y_1,), (\mathbf{\Psi}(\mathbf{x_2}), y_2, ..., (\mathbf{\Psi}(\mathbf{x_n}), y_n,))\} \end{aligned} \tag{2.13}$$

2.2.1.5 Funções *Kernel*

O problema de classificação para o SVM não linear apresentado na Subseção 2.2.1.4 resume-se em uma equação que depende de produtos escalares de funções Ψ no domínio do espaço de características. Neste ponto então, são introduzidas as funções *Kernel*, as quais recebem dois pontos $\mathbf{x_i}$ e $\mathbf{x_j}$ do espaço de entrada S e computam o produto escalar $\mathbf{\Psi}(\mathbf{x_i}) \cdot \mathbf{\Psi}(\mathbf{x_j})$. Os *Kernel* mais comuns são apresentados na Tabela 1.

2.2.2 L-SVM

Em muitas aplicações, as informações rotuladas dos pares de entrada e saída não são suficientes para determinar os relacionamento entre as entradas e saídas de um conjunto, pois esse conjunto depende também de variáveis não observadas ou variáveis sem rótulo, denominadas variáveis latentes.

A fim de generalizar a estrutura das SVMs para esse novo modelo, adapta-se a Equação 2.1 estendendo uma nova variável z, deixando explícito que a solução da nova Equação 2.14 é aquela que maximiza seus valores.

$$f_\beta(\mathbf{x}) = \max_{(z)\epsilon Z(x)} \beta \cdot \mathbf{\Phi}(x, z), \tag{2.14}$$

onde z define os valores latentes possíveis para um exemplo x, e $\mathbf{\Phi}(x, z)$ corresponde a descrição do relacionamento existente entre a entrada x, a saída y e a variável latente z.

Considere que Z_p especifica uma variável latente para cada exemplo de entrada de treinamento x de S. Definindo uma função objetivo auxiliar $L_S(\beta, Z_p) = L_{S(Z_p)}(\beta)$, onde $S(Z_p)$ é derivado de S restringindo os valores latentes segundo Z_p. Ou seja, para um exemplo x_i rotulado de treinamento em S, tem-se $Z(x_i) = \{z_i\}$, onde z_i é o valor latente especificado para x_i por Z_p. Assim, justifica-se usar um conjunto S de treinamento

que contenha entidades não rotuladas em y pela minimização da função auxiliar $L_{S(Z_p)}$ (Equação 2.15).

$$L_S(\beta, Z_p) = \min_{Z_p} L_{S(Z_p)}(\beta) \tag{2.15}$$

Existem alguns métodos que podem resolver a minimização da equação 2.15, como é o caso do algoritmo conhecido como Procedimento Côncavo-Convexo (CCCP) (YUILLE; RANGARAJAN, 2003), que procura minimizar a equação a cada iteração, procurando chegar a um valor mínimo, e outros casos que buscam os melhores valores de *score* inferindo respostas para cada variável latente ou para a variável β (FELZENSZWALB; MCALLESTER; RAMANAN, 2008).

2.3 Detecção de Pedestres por Mistura de Modelos baseados em Partes Deformáveis

A Mistura de Modelos baseados em Partes Deformáveis (MDPM) é um sistema baseado em aprendizado para localização e detecção de objetos em imagens digitais. Essa técnica, apresentada em (FELZENSZWALB; MCALLESTER; RAMANAN, 2008) pode ser caracterizada como uma evolução do detector HOG + SVM (Subseção 2.1) apresentado por (DALAL; TRIGGS, 2005). As melhorias realizadas estão relacionadas sobre: os termos dos filtros aplicados sobre partes; o modelamento das partes (definido como modelo baseado em partes de estruturas em estrela, do inglês, *star-structured part-based models*); e a utilização de valores latentes que caracterizam uma nova versão da SVM conhecida como L-SVM (*Latent SVM*) (Subeção 2.2.2). As variáveis latentes (z) representam parâmetros que constituem partes do corpo humano que não são anotadas explicitamente nas imagens, mas que são estimadas a partir das anotações. A importância do uso das variáveis latentes é que, embora o corpo humano não tenha uma estrutura rígida, o que dificulta a sua detecção, ela pode ser decomposta em regiões menores aproximadamente rígidas, e essas regiões são modeladas esperançosamente pelas variáveis latentes. As *boxes* azuis na Figura 11 apresentam a configuração de partes em uma imagem contendo pessoas.

Para se obter o MDPM são necessários os seguintes passos:

- Extrair as características de HOG da imagem digital;
- Destacar os "filtros raízes" e "filtros partes", e definir os componentes do Modelo de Partes Deformáveis;
- Construir a pirâmide de carcterísticas;
- Realizar o processo de *matching* calculando os *scores* de contribuição de cada filtro;

- Aprendizado dos parâmetros pela L-SVM.

Nas próximas subseções os passos acima são detalhados para caracterizar a detecção de pedestres.

2.3.1 Filtros e Pirâmide de Características

Os modelos deformáveis são construídos a partir das características HOG (Subseção 2.1) e por operações de filtragem. Neste caso, um filtro é um padrão retangular definido por um mapa de características HOG. O termo filtro é utilizado pois as características do HOG representam pesos para o método de detecção.

Uma pirâmide de mapas de características de HOG, definida como $H = (H_1, ..., H_l)$ é criada por borramento e subamostragem de uma imagem digital calculando o mapa de características HOG para cada nível (l_i) da pirâmide. São definidos em H dois filtros chamados *filtro raiz* e *filtro de partes*. O *filtro raiz* refere-se à representação bruta (*coarse*) dos objetos, enquanto que os *filtros de partes* referem-se a uma representação mais detalhada do objeto, pois é obtido com o dobro da resolução no qual o *filtro raiz* foi obtido. Os *filtros de partes* são calculados ao descer λ degraus na pirâmide, partindo do nível l_0, nível do *filtro raiz*, e avançam nos demais níveis $l_i = l_0 - \lambda$ onde os *filtros partes* são calculados com o dobro de resolução. Usualmente, λ=5 na etapa de treinamento e λ=10 na etapa de teste (FELZENSZWALB; MCALLESTER; RAMANAN, 2008). A Figura 12 exemplifica a pirâmide de características de uma imagem contendo o *filtro raiz* definido pela janela de detecção na cor ciano em l_0 e *filtros de partes* localizado a λ níveis baixo na pirâmide.

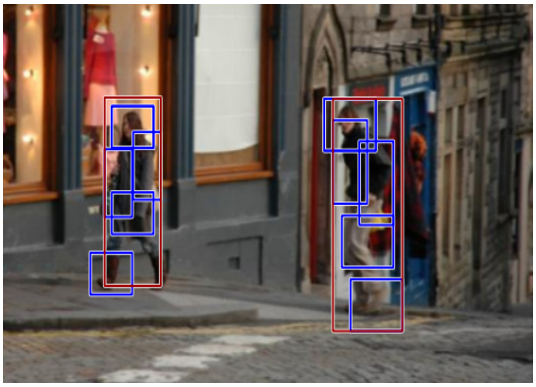


Figura 11 – Exemplo de modelo indicando divisão de pessoas que devem ter suas características extraídas. Essas partes não possuem rótulo nas anotações (variáveis latentes).

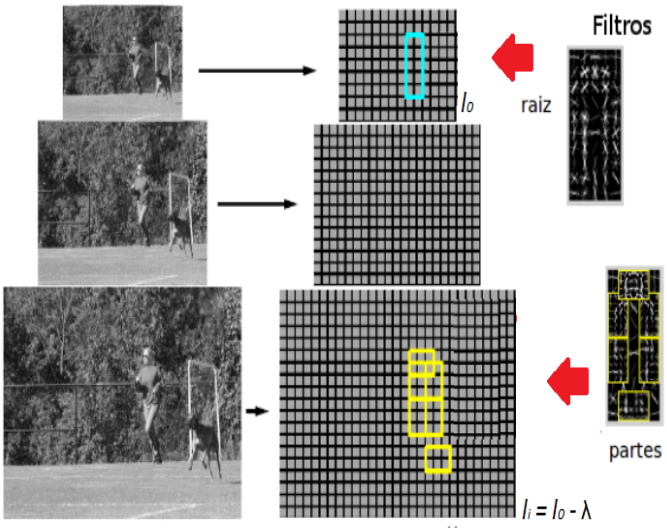


Figura 12 – Uma pirâmide de imagens e uma parametrização do modelo de uma pessoa dentro desta pirâmide. Os *filtros parte* são estimados ao dobro da resolução espacial do *filtro raíz*. (Adaptada de (FELZENSZWALB; MCALLESTER; RAMANAN, 2008)

2.3.2 Modelos de Parte Deformável

Seja M uma mistura de modelos definida por m componentes $(M_1, ..., M_m)$. O modelo para a c-ésima componente (M_c) é baseado em um modelo estrela de uma estrutura pictórica, ou seja, ele é definido por filtros lineares (*filtro raiz* e *filtros de partes*) e um custo de deformação para cada parte. Formalmente, o modelo da c-ésima componente da mistura formada por n partes consiste de uma $(n+2)$-tupla (Equação 2.16). Como exemplo, considere que se deseja construir o modelo de uma pessoa, o *filtro raiz* representa os limites estabelecidos pela localização da pessoa, e as n partes representam detalhes dessa pessoa como cabeça, membros superiores e membros inferiores.

$$M_c = (F_0, P_1, ..., P_n, b), \tag{2.16}$$

onde F_0 é o *filtro raiz*, n representa a quantidade de partes, P_i $(i = 1, 2, ..., n)$ é o *modelo de parte* i e a variável b é um valor real que representa um viés. A Figura 13 exemplifica a imagem de uma pessoa com uma quantidade de $n = 5$ partes. Cada *modelo de parte* é definido por uma 3-tupla $P_i = (F_i, v_i, d_i)$, sendo F_i o i-ésimo *filtro de parte*, v_i é um vetor de duas dimensões que representa a localização fixa (*anchor position*) da parte i em relação à posição do *filtro raiz*, e d_i é um vetor de 4 dimensões que especifica os coeficientes de um função custo de deformação quadrática de uma parte i em relação a v_i. Em seguida um detalhamento maior sobre a definição dos vetores v_i e d_i da 3-tupla (F_i, v_i, d_i):

- v_i é um vetor bidimensional que indica uma localização fixa da parte i em relação

à posição da raiz, tida como o canto superior esquerdo da janela do *filtro raiz*. Por exemplo, na Figura 13c temos o vetor v_3, de cor ciano, que indica a localização fixa do *filtro de parte* F_3 com respeito ao *filtro raiz*.

- d_i é um vetor quadridimensional que indica os parâmetros ou coeficientes de uma função quadrática que define um custo de deformação para cada localização i dos *filtros de partes* em relação à v_i. Da Figura 13d observa-se o deslocamento das partes em um processo de casamento (*matching*) entre o modelo e uma instância de objeto, no caso, uma pessoa em uma imagem. Neste caso, as características de deformação são dadas pela função quadrática: $\phi_d(dx_i, dy_i) = (dx, dy, dx^2, dy^2)$, onde $(dx_i, dy_i) = (x_i, y_i) - (2(x_0, y_0) + v_i)$ representa o deslocamento da parte. Logo, se por exemplo, os coeficientes da função quadrática, d_i forem (0,0,1,1), então temos que o custo de deformação da parte i é o quadrado da distância entre a posição atual e a posição fixa em relação ao *filtro raiz*.

Cada hipótese objeto (variável latente) $z = (p_0, ..., p_n)$ especifica uma componente da mistura M_c, $(1 \leqslant c \leqslant m)$, e a localização para cada filtro da componente na pirâmide de características H, onde $p_i = (x_i, y_i, l_i)$ especifica a posição (x_i, y_i) e o nível l_i do i-ésimo filtro na pirâmide H.

2.3.3 Processo de *Matching*

Em (FELZENSZWALB; MCALLESTER; RAMANAN, 2008) é abordado o problema de detecção em imagens com uma análise de *matching* por janelas deslizantes de misturas de modelos multi-escalares de partes deformáveis (MDPMs). Para o *matching*

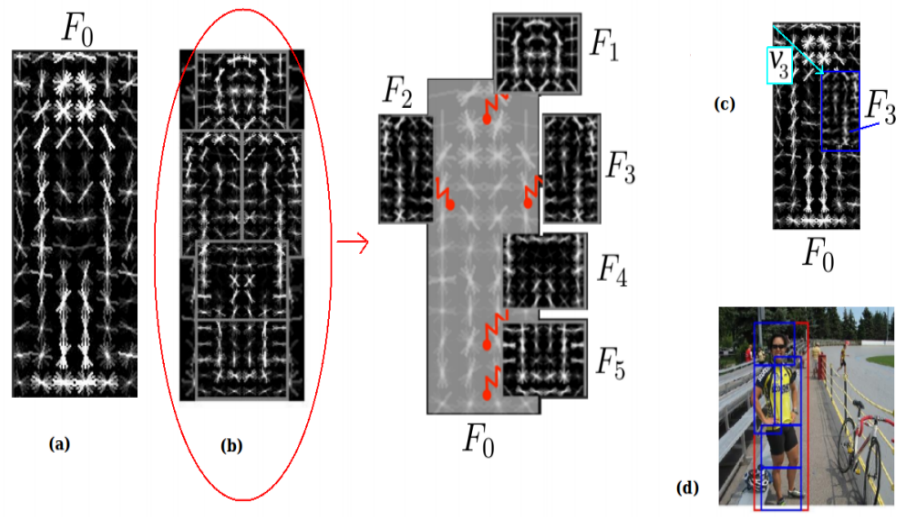


Figura 13 – *Filtros raiz* e *filtros de partes* sobre a pirâmide de características HOG. (Adaptada de (FELZENSZWALB; MCALLESTER; RAMANAN, 2008)

de uma estrutura pictórica-estrela com uma imagem, usa-se programação dinâmica, ou seja, que se adapta aos resultados para encontrar as localizações mais eficientes das partes como uma função da localização dos *filtros raízes*. Esse processo de *matching* consiste em definir uma pontuação (*score*) para cada localização do *filtro raiz* de componente.

A Figura 14 ilustra o processo de *matching*, sendo possível verificar que as respostas dos *filtros raízes* e *filtros partes* são calculadas em diferentes resoluções na *pirâmide de características*. As respostas transformadas são combinadas para encontrar um *score* final para cada localização de raiz. Esta figura apresenta as respostas e suas transformações para as partes de "cabeça" e "ombro direito". O resultado da combinação das respostas mostram duas hipóteses visualmente distinguíveis pela legenda de cores apresentada.

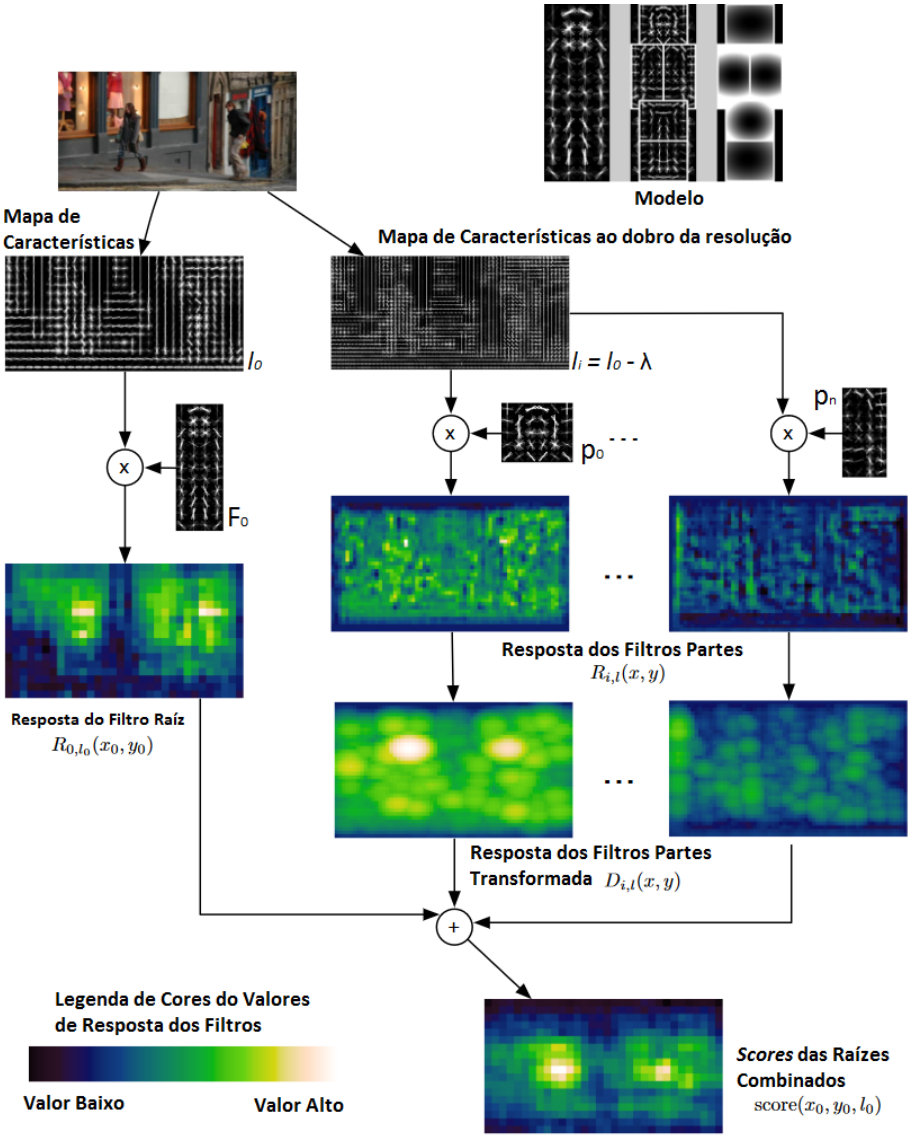


Figura 14 – Processo de *matching*. (Adaptada de (FELZENSZWALB; MCALLESTER; RAMANAN, 2008)

Um alto *score* de localização da raiz define uma detecção. O processo de detecção, ou *matching*, é dado pela Equação 2.17.

$$score(p_0) = \max_{p_i,\ldots,p_{n_c}} score(z). \tag{2.17}$$

Assim, a pontuação da hipótese objeto, $score(z)$ (Equação 2.18), é dada pela pontuação dos filtros, e suas localizações, menos um custo de deformação que depende da posição relativa de cada parte em relação ao *filtro raiz*, mais um valor real chamado viés (b) inserido na Equação 2.18 para possibilitar a comparação de *scores* múltiplos quando as partes são combinadas em um *modelo de misturas*.

$$score(z) = \sum_{i=0}^{n_c} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n_c} d_i \cdot \phi_d(dx_i, dy_i) + b, \tag{2.18}$$

onde $\phi(H, p_i)$ é um vetor obtido pela concatenação dos vetores de características na sub-janela $w \times h$ de H com o canto superior esquerdo em $p_i = (x_i, y_i, l_i)$ ordenado em ordem decrescente em linha. $\phi(dx_i, dy_i) = (dx, dy, dx^2, dy^2)$ é uma função quadrática das características de deformação, e d_i são os coeficientes de uma função quadrática.

Considere $R_{i,l}(x, y) = F_i \cdot \phi(H, p_i)$ um vetor que contém a resposta do i-ésimo modelo de filtro no nível l da pirâmide de características H. O algoritmo de *matching* inicia-se pelo cálculo dessa respostas. Observe que $R_{i,l}$ é uma correlação cruzada entre F_i e o nível l da pirâmide de características.

Uma vez calculado as respostas dos filtros por $R_{i,l}(x, y)$, essas respostas passam por uma transformação que espalha *scores* de alta frequência nas localizações próximas, levando em conta o custo de deformação pela Equação 2.19.

$$D_{i,l}(x, y) = \max_{dx,dy}(R_{i,l}(x + dx, y + dy) - d_i\phi_d(dx, dy)). \tag{2.19}$$

O valor de $D_{i,l}(x, y)$ representa a máxima contribuição da i-ésima parte para o *score* de uma localização de raiz "ancorada" na posição (x, y) no nível l. Logo, a pontuação de localização do filtro raiz, definida na Equação 2.17, pode ser reescrita representando o *score* geral dos *filtros raízes* (Equação 2.20), sendo representada pela soma das respostas dos *filtros raízes* naquele nível, mais as versões deslocadas das transformadas e respostas das partes subamostradas.

$$score(p_0) = R_{0,l_0}(x_0, y_0) + \sum_{i=1}^{n_c} D_{i,l_0-\lambda}(2(x_0, y_0) + v_i) + b. \tag{2.20}$$

2.3.4 Classificação

O processo de classificação neste trabalho refere-se ao treinamento dos classificadores sobre a estrutura da MDPM, dos filtros e os custos de deformação, e o uso das regras aprendidas para a detecção ou não de pedestres. Neste processo, os algoritmos de SVM Linear (Subseção 2.2.1) e L-SVM (Subseção 2.2.2) podem ser utilizados.

Ao utilizar a SVM,em (DALAL; TRIGGS, 2005) é utilizado *kernel* gaussiano, enquanto que (FELZENSZWALB; MCALLESTER; RAMANAN, 2008) faz uso de *kernel* quadrático, vide seção 2.2.1.5. O classificador basicamente determina se existe ou não um instância de pessoa em uma dada posição e escala da imagem. Para construção do classificador LSVM (Seção 2.2.2), neste modelo baseado em partes, β é tido como um vetor de filtros raízes, filtros de partes e pesos de custo de deformação, enquanto z especifica a configuração das pessoas, sendo $\Phi(x, z)$ a concatenação de subjanelas referentes à piramide de característica e deformação das partes.

2.3.4.1 L-SVM para as MDPMs

Para se realizar o aprendizado dos parâmetros das componentes MDPM pelas SVMs Latentes, a Equação 2.18 deve ser redefinida da seguinte forma, onde $\beta \cdot \Phi(H, z)$ representa a operação de produto escalar entre β e $\Phi(H, z)$:

$$\begin{aligned} score(z) &= \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b, \\ &= \underbrace{(F_0, ..., F_n, d_1, ..., d_n, b)}_{\beta} \cdot \underbrace{(\phi(H, p_0), ..., \phi(H, p_n), -\phi_d(dx_1, dy_1), ..., -\phi(dx_n, dy_n), 1)}_{\Phi(H,z)}, \\ &= \beta \cdot \Phi(H, z). \end{aligned}$$

Assim, na formulação do SVM Latente, a função linear que se deseja aprender, partindo da definição dada pela Equação 2.14 (Subseção 2.2.2), fica da seguinte forma:

$$f_\beta(\mathbf{x}) = \max_{z \epsilon Z(x)} \beta \cdot \mathbf{\Phi}(H, z). \tag{2.21}$$

2.3.5 Aprendizagem de Parâmetros

Assume-se que os exemplos de treino são dados por um conjunto P de pares (I, B), onde I é uma imagem e B *bounding boxes* representando os exemplos positivos, e um conjunto N de imagens de *background*.

Seja M um conjunto de modelo de misturas dado pelos parâmetros definidos em β da (Equação 2.21), o qual representa os F_i filtros. Para aprender β o problema é definido como um treinamento de SVM Latente com um conjunto D que contém P e N.

Cada exemplo x pertencente à D possui uma imagem e uma pirâmide de características H associada. Os valores Latentes de $z \epsilon Z(x)$ especificam uma instância de M na pirâmide de características $H(x)$. Assim, a Equação 2.21 representa exatamente o *score* da hipótese z para M em $H(x)$.

Para a obtenção das MDPMs, faz-se uso dos exemplos positivos e negativos, e são separados em algumas fases. Os exemplos positivos são ordenados após o cálculo das *relações de aspecto* das *bounding boxes* das imagens, as quais são calculadas pela razão entre a largura w e a altura h das *bounding boxes*. A ordenação garante que os exemplos positivos fiquem divididos em m grupos de mesmo tamanho $(P_1, ..., P_m)$, sendo m a quantidade de componentes da mistura.

Para este trabalho, o detector foi treinado na base de dados INRIA[1], a qual é formada por imagens com exemplos positivos, que contém somente uma pessoa isolada por imagem, e imagens com exemplos negativos, isto é, imagens sem a presença de pessoas. As detecções realizadas em um *frame* I^k resultam em um conjunto $D_k = \left\{d_1^k, ..., d_{n_k}^k\right\}$, onde d_j^k é a j-ésima detecção em D_k, e cada entidade d_j^k representa a localização e as dimensões das *boxes* detectadas, dado por $d_j^k = \left\{x_j^k, y_j^k, w_j^k, h_j^k, score_j^k\right\}$. A título de padronização, o par x_j^k e y_j^k corresponde à localização do canto esquerdo superior da *box* j, w_j^k e h_j^k são a largura e altura da *box* j, respectivamente. O último termo, $score_j^k$, refere-se a avaliação daquela detecção calculada pelo detector. Quanto maior o valor de $score_j^k$ mais confiável foi aquela detecção.

Em termos práticos, a saída do detector fornece uma série de *boxes*, representando cadidatos a pessoas, com seus respectivos *scores*. A fim de obter uma maior assertividade sobre esses resultados, um limiar ajustável é imposto sobre os *scores*, *scores* menores do que um limiar t (limiar de *score*) são eliminados. Um outro limiar B, conhecido como limiar de sobreposição de hipóteses, é também imposto às detecções de saída. Devido à multiescalaridade desta técnica, diversas *bounding boxes* podem ser encontradas sobre uma mesma pessoa, sendo então necessário eliminar tais redundâncias. Dessa forma detecções com sobreposições maiores do que B são indicadas como "falsos positivos" e então descartadas. Os valores de t e B utilizados neste trabalho estão disponíveis no Capítulo 5.

2.4 Detecção de Pedestres por Descorrelação Local de Canais de Características - LDCF

A técnica de Detecção de Pessoas por Descorrelação Local de Canais de Características (do inglês, *Local Decorrelation Channel Features* - LDCF), proposta em (NAM;

[1] *http://lear.inrialpes.fr/data*

DOLLAR; HAN, 2014), é inspirada em trabalhos que trouxeram a abordagem de descorrelação global de características de HOG como aquele em (HARIHARAN; MALIK; RAMANAN, 2012). Porém, na LDCF é utilizada descorrelação local que remove correlação entre canais de características calculadas para cada imagem, e que se mostrou mais eficiente segundo (NAM; DOLLAR; HAN, 2014). Os resultados encontrados são representações de características descorrelacionadas e que são utilizadas com árvores de decisão ortogonal para a detecção de pedestres. Neste trabalho, entende-se descorrelação como um termo geral para exemplificar um processo usado para reduzir a autocorrelação entre entidades.

Como *baseline* para a aquisição das características na LDCF é utilizada a abordagem de Canais de Características Agregadas (do inglês, *Aggregated Channel Features - ACF*), vista em (DOLLAR et al., 2014) e descrita na Subseção 2.4.2.

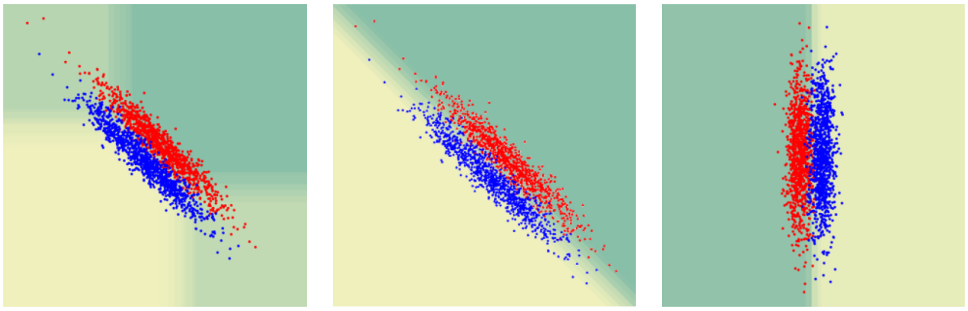
2.4.1 Árvores Ortogonais Impulsionadas

Impulsionamento é uma simples ferramenta, embora poderosa em termos de classificação, que pode modelar funções não lineares complexas. A ideia é basicamente treinar e combinar um número de "classificadores fracos" em um classificador mais preciso. As árvores de decisão são frequentemente utilizadas como "classificadores fracos" em conjunto com a técnica de impulsionamento.

Tipicamente, árvores de decisão impulsionadas são treinadas com divisões ortogonais (características únicas); no entanto, a extensão para divisões oblíquas (multi-características) são comuns e amplamente utilizadas em classificadores, como em (MENZE et al., 2011) e (RODRIGUEZ; KUNCHEVA; ALONSO, 2006). Em (NAM; DOLLAR; HAN, 2014) é visto que uma transformação de descorrelação local de características pode eliminar a necessidade de divisões oblíquas e se mostra mais eficiente.

A Figura 15 mostra o resultado de classificação de um mesmo conjunto de dados pelas técnicas de classificação mencionadas acima. Verifique que a classificação por árvore ortogonal impulsionado com características descorrelacionadas (Figura 15c) separa satisfatoriamente os dados, assim como a aplicação da árvore oblíqua na Figura 15b.

Algumas definições são importantes: 1) Árvores de decisão com divisões oblíquas podem modelar de maneira mais eficiente dados com características correlacionadas, 2) Em árvores oblíquas, toda divisão é baseada em uma projeção linear do tipo: $z = \mathbf{w^T x}$, onde $\mathbf{w}$ representa uma projeção e pode ser encontrada por análises discriminantes lineares (do inglês, *linear discriminant analysis* - LDA), e $\mathbf{x}$ é uma entrada. A LDA busca minimizar o espalhamento intra-classes, enquanto maximiza o espalhamento inter-classes (JAIN; DUIN; MAO, 2000).



(a) Árvore Ortogonal. (b) Árvores Oblíqua. (c) Árvore Ortogonal Impulsionada com características descorrelacionadas.

Figura 15 – Saída de técnicas de classificação por árvores de decisão. (Adaptado de (NAM; DOLLAR; HAN, 2014))

2.4.2 Canal de Características Agregadas - ACF

Dada uma imagem de entrada I, a técnica ACF calcula diversos canais de características $C = \Omega(I)$, sendo cada canal um mapa de características encontrado pelos *pixels* da imagem. Os canais de características utilizados na ACF são os mesmo encontrados em (DOLLAR et al., 2014): Gradientes de Magnitude Normalizado (1 canal), Histogramas Orientados à Gradientes (6 canais), e canais no espaço de cores LUV (3 canais), totalizando 10 canais. Antes do cálculo dos 10 canais de características, a imagem de entrada I é suavizada por um filtro $[1 \quad 2 \quad 1]/4$. Os canais são então divididos em blocos de 4×4 canais e os *pixels* em cada bloco são somados (agregação). Ao final, os canais são novamente suavizados por um filtro de mesmas propriedades $[1 \quad 2 \quad 1]/4$.

Os canais são subamostrados em duas vezes, sendo as características resumidas aos valores dos canais encontrados para cada *pixel*.

O treinamento do modelo é realizado pela técnica apresentada em (FRIEDMAN; HASTIE; TIBSHIRANI, 1998) com múltiplas rodadas para treinar e combinar 2048 árvores de decisão com profundidade 3 sobre as características para distinguir os objetos do *background*.

A Figura 16 apresenta o fluxo básico de aplicação do detector ACF proposto em (DOLLAR et al., 2014).

2.4.3 Canais de Características Localmente Descorrelacionadas - LDCF

A utilização dos Canais de Características Localmente Descorrelacionadas partem de algumas observações realizadas em (NAM; DOLLAR; HAN, 2014):

1. Divisões Oblíquas aprendidas com LDA (*linear discriminant analysis*) sobre os $m \times m$

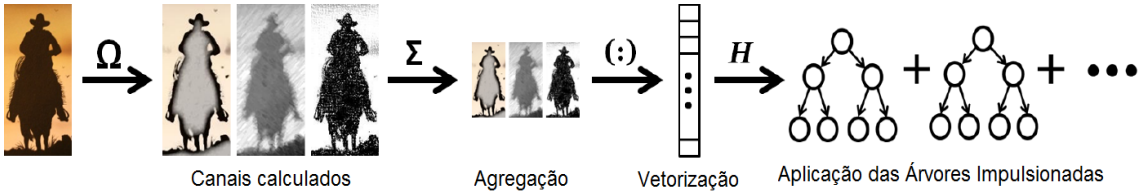


Figura 16 – *Overview* do Detector ACF. (Adaptado de (DOLLAR et al., 2014))

pacotes locais aprimoram os resultados sobre as divisões ortogonais. Os pacotes são partes menores I^k de uma imagem completa, considerando que uma imagem I pode ser decomposta em entidades menores ($I = [I^1...I^k]$). O cálculo dos pacotes pode ser encontrado em (DOLLAR et al., 2014).

2. Uma matriz de Covariância $\sum$ pode ser compartilhada entre todos os pacotes de uma imagem I.

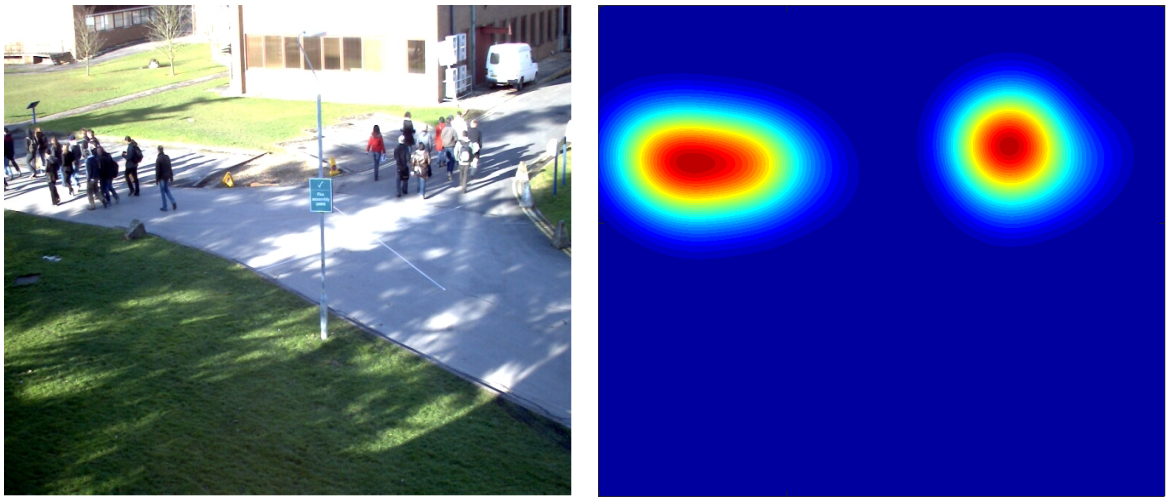
3. Árvores ortogonais com características descorrelacionadas podem potencialmente ser usadas no lugar de árvores oblíquas.

Em resumo, a LDCF modifica a ACF utilizando os 10 canais e aplica $k = 4$ filtros lineares de descorrelação por canal. Os filtros de descorrelação são estimados pela técnica descrita em (NAM; DOLLAR; HAN, 2014). O resultado é um conjunto de 40 "características de canais descorrelacionados localmente"(LDCF) que seguem todos os mesmos passos de treinamento e testes para o ACF, conforme mencionado na Subseção 2.4.2.

3 Mapa de Densidade

O uso de mapas de densidade auxilia na identificação das regiões mais prováveis de presença de pessoas, sendo portanto um método para melhorar a precisão de um sistema de detecção de pedestres.

Baseado em (FRADI; DUGELAY, 2015), neste trabalho é utilizado uma metodologia que prevê a extração de características locais da imagem indicando a presença do conteúdo de interesse. Em seguida, rastreia-se o conteúdo encontrado aplicando fluxo óptico, o qual é responsável por eliminar características desprezíveis, como regiões que aparecem e desaparecem na imagem, por um processo conhecido como *forward-backward*. Ao final desse processo, o mapa de densidade de pessoas é calculado por uma função de soma de gaussianas fornecendo os valores de contribuição da densidade por *pixel* da imagem. O resultado desse processo pode ser visto na Figura 17.



(a) Imagem com grupo de pessoas

(b) Mapa de Densidade da Figura 17a

Figura 17 – Mapa de Densidade Visual de uma imagem da base de dados PETS2009.[2]

O presente capítulo tem por objetivo apresentar a metodologia utilizada *frame* a *frame* para calcular os mapas de densidade deste trabalho. Para isso, inicialmente são apresentadas técnicas de extração de características de imagens digitais na Subseção 3.1 que podem ser utilizadas em conjunto com o fluxo óptico (Subseção 3.2) para detecção de movimento e construção do mapa. Uma subseção adicional é ainda apresentada (Subseção 3.3) para exploração de um passo relevante conhecido como *forward-backward* para refinar a saída da etapa de fluxo óptico. Por fim, a integração dessas técnicas são compiladas na Subseção 3.4 com o detalhamento de como desenvolver os mapas de densidade.

[2] http://www.cvg.reading.ac.uk/PETS2009/

3.1 Extratores de Características

Características são informações extraídas de um conjunto de dados, e que representam de forma adequada esse conjunto. Em processamento de imagens, utiliza-se extração de características para representar de forma eficiente pontos de interesse de uma imagem. Esses pontos ou características de interesse podem ser bordas, cantos ou curvaturas, e fazem parte de aplicações como detecção e classificação de objetos, rastreamento, recuperação de conteúdo, percepção de movimento e classificação de textura. Em imagens, características estão geralmente associadas a porções da imagens que diferem de seu arredor por textura, cor ou intensidade (DAVIS, 1975). As seções a seguir apresentam algumas das técnicas de extração de características que são usadas neste trabalho.

3.1.1 SIFT

A técnica SIFT (Transformada de Características Invariantes à Escala, do inglês, *Scale-Invariante Feature Transform*), proposta por (LOWE, 1999), é um algoritmo utilizado em imagens digitais para o reconhecimento de características locais. Os descritores SIFT de imagens provêm um grupo de características que são invariantes à rotação, translação e robustas à mudança de escala, iluminação e a pequenas faixas de projeção. Assim, ao mesmo tempo que é possível reconhecer características SIFT em uma mesma imagem redimensionada, essas características também são encontradas caso a imagem seja observada em diferente posições em um mesmo ambiente. As características SIFT também são muito robustas quanto ao efeito de ruídos na imagem.

3.1.1.1 Detecção de Máximo/Mínimo no Espaço de Escalas

A fim de alcançar invariância em escala, SIFT utiliza uma função DoG (Diferença de Gaussianas, do inglês, *Difference of Gaussians*), apresentada na Equação 3.1, para realizar convolução em uma imagem. A variação do valor de σ na Equação 3.2 permite a obtenção da imagem em diferentes escalas.

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y), \\ &= L(x, y, k\sigma) - L(x, y, \sigma), \end{aligned} \tag{3.1}$$

onde:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3.2}$$

onde $I(x, y)$ representa uma imagem de entrada, com suas coordenadas de *pixel* (x, y), e k representa um coeficiente de escala de um fator de espaçamento de escala adjacente. A partir do cálculo de $D(x, y, \sigma)$ as *keypoints* são identificadas nas posições de local

mínima/máxima nas imagens DoG. Esse procedimento é realizado pela comparação de cada *pixel* nas imagens DoG com seus 26 *pixels* vizinhos adjacentes. Os 26 *pixels* correspondem a oito vizinhos na mesma escala $k\sigma$, mais os outros nove vizinhos uma escala acima ($k\sigma$+1) e uma escala abaixo ($k\sigma$-1). Se o valor de D analisado é um mínimo ou um máximo entre todos os pontos, então essa localização do ponto e escala são armazenados. A Figura 18 exemplifica o posicionamento dos 26 vizinhos adjacentes do *pixel* em análise representado por um "X" na figura.

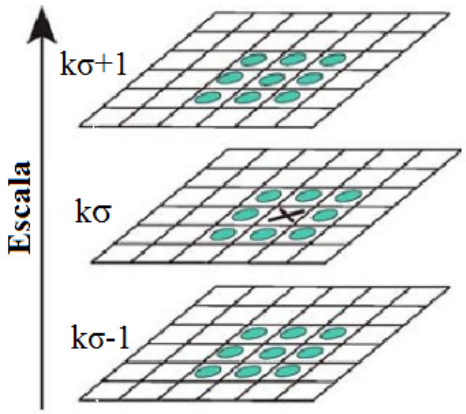


Figura 18 – Visualização dos 26 vizinhos adjacentes na busca dos *keypoints* da SIFT.

3.1.1.2 Localização de *Keypoints*

Em seguida, um passo de remoção de *keypoints* de baixo contraste é realizado pelo cálculo da expansão de segunda ordem de Taylor de $D(x, y, \sigma)$, no qual *keypoints* que apresentarem respostas com valores calculados menores que um determinado *offset* são descartados. Ainda, há uma etapa de eliminação de *keypoints* que possuem resposta não significativa para bordas (*edges*) ao utilizar autovalores de segunda ordem da matriz Hessiana 2×2.

3.1.1.3 Atribuição de Orientação

Essa etapa tem por objetivo atribuir uma orientação consistente para os *keypoints* baseados em sua localização na imagem. Calcula-se, na escala de cada *keypoint*, o gradiente de magnitude e orientação de cada vizinho ao seu redor pelas Equações 3.3 e 3.4.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \tag{3.3}$$

$$\theta(x, y) = tan^{-1}\left(\frac{(L(x,y+1)-L(x,y-1))}{(L(x+1,y)-L(x-1,y))}\right) \tag{3.4}$$

A partir das magnitude e orientações, para a região ao redor de cada *keypoint*:

- É criado um histograma com 36 orientações diferentes, cada orientação defasada de 10°, sendo que cada amostra adicionada no histograma é balanceada pela sua magnitude e por uma janela gaussiana circular ponderada com um σ que é 1,5 vezes a escala do *keypoint* em questão.
- O valor máximo do histograma irá representar sua orientação.
- O valor máximo então é interpolado usando uma parábola em todo os três maiores picos mais próximos.
- No histograma, para cada orientação acima de 80% do pico, um novo *keypoint* é criado, com mesma escala e posição, mas orientação diferente.

A Figura 19 exemplifica um procedimento onde foram calculadas as magnitudes e orientações para uma imagem, e o histograma das diferentes orientações, apresentando dois picos acima de 80%.

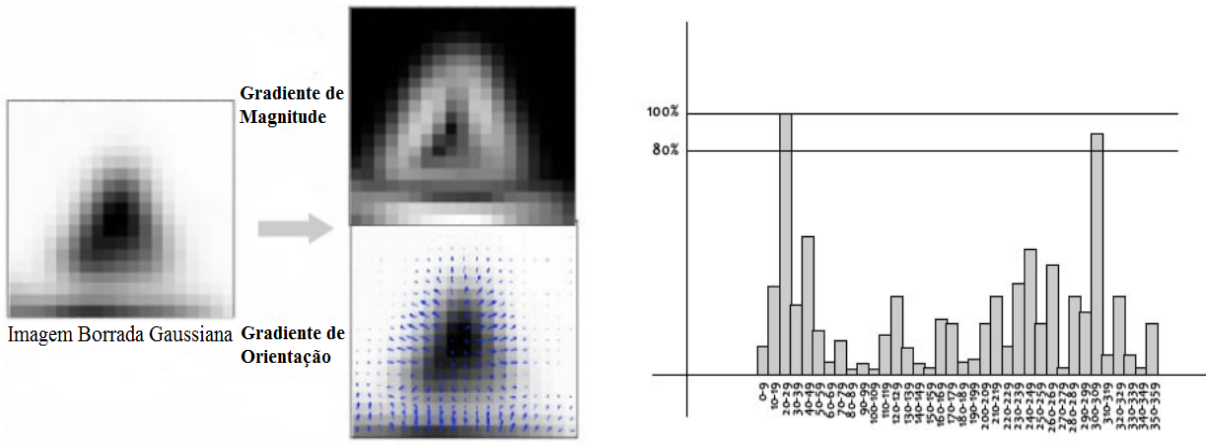


Figura 19 – Magnitude, Orientação e Histograma em um processo SIFT.

3.1.1.4 Descritor de Características

Até aqui foram encontrados as localizações, escalas e orientações, o que garante sua invariância a essas variáveis. Agora, por fim, deseja-se criar uma impressão digital, ou descritor, para as características a partir dos *keypoints* selecionados.

Para isso, uma janela de 16×16 *pixels* ao redor do *keypoint* é selecionada. Essa janela então é dividida em dezesseis janelas menores de 4×4. Dentre as janelas 4×4 são calculados gradientes de magnitude e orientação. Esses gradientes são então agrupados em um histograma de 8 divisões. Ao somar uma orientação ao histograma, a magnitude

é ponderada por uma gaussiana, de forma que as orientações referentes a pontos mais distantes da localização do *keypoint* têm menor peso.

O descritor de cada ponto tem ao final um vetor de 128 características (8 orientações $\times 4 \times 4$ janelas) para reduzir o efeito de interferência devido a iluminação, o vetor resultante que define o descritor é normalizado aplicando-se um limiar de 0,2, então normaliza-se novamente o vetor.

3.1.2 ASIFT

ASIFT (Transformada Afim de Características Invariantes à Escala, do inglês, *Affine Scale-Invariant Feature Transform*) é um extrator de características invariantes afins baseado na técnica SIFT (Seção 3.1.1), que além da invariância quanto à escala, rotação e translação, também trata de modificações nos eixos de orientação da câmera.

3.1.2.1 Transformada Afim

Uma Transformação Afim de uma imagem $I_1(x_1, y_1)$ é um mapeamento que leva essa imagem a uma outra imagem $I_2(x_2, y_2)$ seguindo a seguinte equação:

$$I_2(x_2, y_2) = A * I_1(x_1, y_1) + T, \tag{3.5}$$

onde A é uma matriz inversível e $T \epsilon R^2$ é o vetor de translação.

A Transformação Afim de uma imagem $I(x, y)$ é uma transformação linear com seis graus de liberdade (2 de escalonamento, 2 de rotação, 2 de translação).

Existem duas propriedades básicas importantes sobre a transformada afim:

- A transformada Afim preserva o paralelismo de retas;
- A transformada Afim preserva a razão de distância entre pontos.

A Figura 20 apresenta variações possíveis de transformadas afins em uma imagem.

3.1.2.2 Decomposição Afim para Diferentes Pontos de Vista

A técnica apresentada em (YU; MOREL, 2011) baseia-se na utilização de parâmetro de orientação da câmera presente na interpretação geométrica do sistema plano objeto e da câmera fotográfica.

Considere a Figura 21 que descreve a interpretação geométrica de uma decomposição afim, onde u é o plano observado pela câmera, ψ é o ângulo de rotação da câmera sobre seu eixo óptico, λ é o parâmetro de escala (distância câmera-ponto central do plano), ϕ

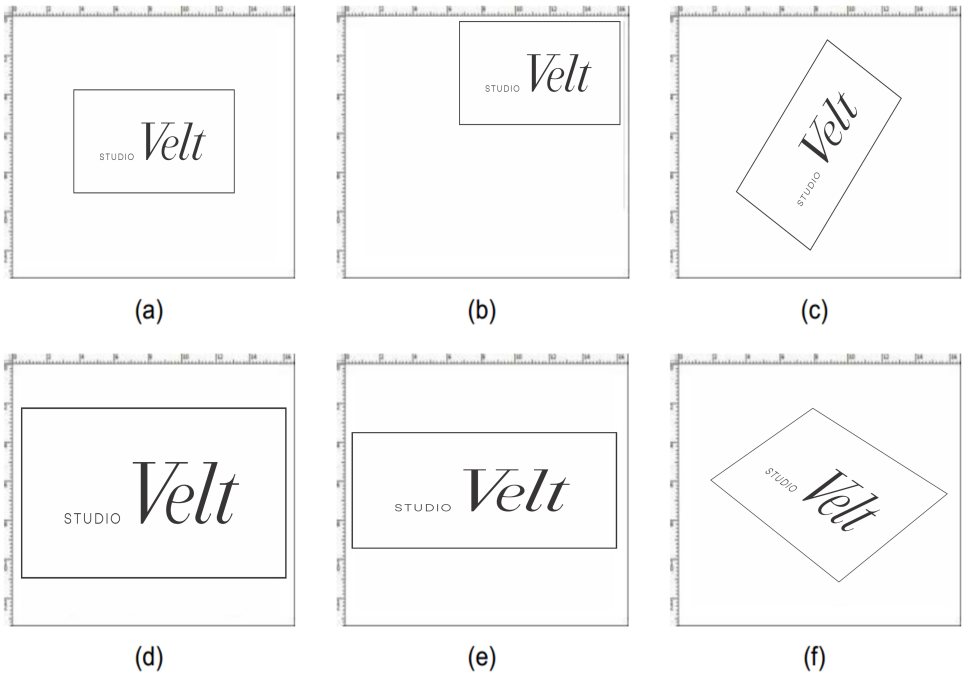


Figura 20 – Transformadas Afins a) Imagem original, b) translação, c) rotação, d) escala uniforme, e) escala não uniforme, e f) combinação de alterações na imagem.

é o ângulo de rotação da câmera sobre o plano da imagem e θ é o ângulo de inclinação medido entre a normal do plano da imagem e o eixo ótico da câmera. Assume-se que a câmera, representada pela imagem de um olho na Figura 21, está distante do plano u e inicia-se em uma visão frontal com parâmetros $\lambda = 1$, $\phi = \psi = 0$.

O mapa A afim possui determinante positivo e equaciona a decomposição apresentada na Figura 21 pela Equação 3.6. Esta matriz A, é aquela matriz inversível definida na Equação 3.5.

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = H_\lambda R_1(\psi) T_t R_2(\phi) = \lambda \begin{bmatrix} cos\psi & -sen\psi \\ sen\psi & cos\psi \end{bmatrix} \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} cos\phi & -sen\phi \\ sen\phi & cos\phi \end{bmatrix} \tag{3.6}$$

onde $\lambda > 0$, λt é o determinante de A, R_i são as rotações, $\phi\epsilon[0, \pi)$, H_λ é a matriz de escala e T_t é a mudança de inclinação com $t = 1/cos\theta$. O parâmetro t, chamado de *tilt* (grau de inclinação), é utilizado com duas definições:

- *tilt* absoluto: representa a diferença de inclinação de uma imagem para a sua vista frontal.
- *tilt* de transição: representa a medida de inclinação entre dois pontos de vista de uma imagem. É importante que o algoritmo de extração de características seja invariante a altos valores de *tilts* de transição.

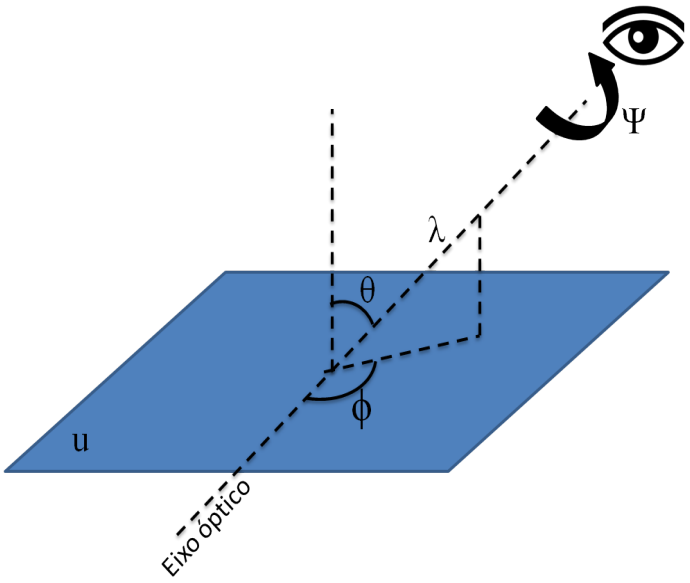


Figura 21 – Interpretação Geométrica da Decomposição Afim.

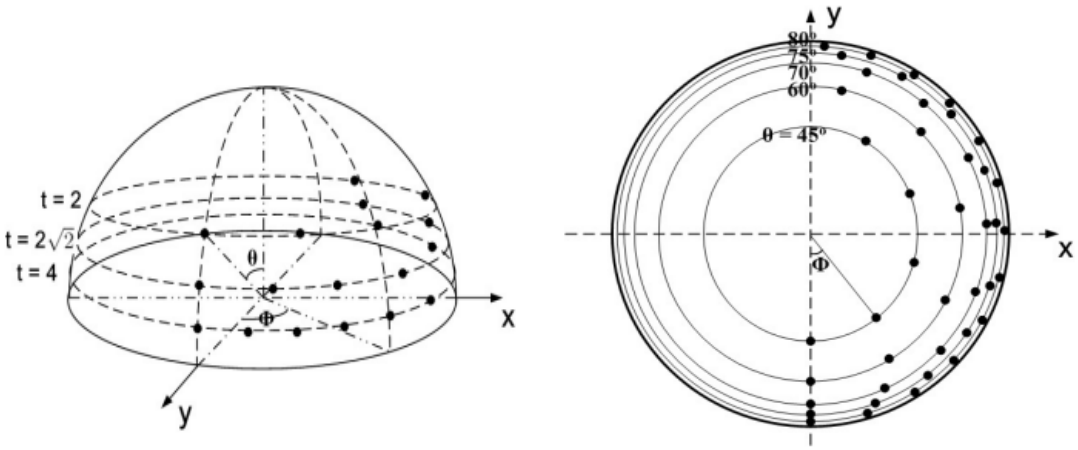
3.1.2.3 Procedimento de Cálculo do ASIFT

1. Cada imagem é transformada simulando todas as possíveis distorções afins causadas pela alteração no eixo óptico da câmera. Essas distorções dependem em especial dos parâmetros de longitude ϕ e latitude θ, que podem ser visualizados na Figura 21. As imagens passam por rotações de ângulo ϕ seguidas por inclinações (*tilts*) com parâmetro $t = 1/|cos\theta|$. Uma inclinação de t na direção x é representada pela operação $u(x, y) \rightarrow u(tx, y)$. As inclinações são realizadas por subamostras de valores de t. Antes das simulações de inclinação, um filtro de *antialiasing* é aplicado na direção x. Este filtro é especificado pela convolução da imagem com uma Gaussiana de desvio padrão $c\sqrt{t^2 - 1}$. O valor $c = 0,8$ é estipulado empiricamente por (YU; MOREL, 2011) e proporciona um erro pequeno de *aliasing*.

   As rotações e inclinações são realizadas para um número finito e pequeno de ângulos de latitudes e longitudes que garantem que as imagens simuladas permaneçam próximas à qualquer outra visão possível gerada por ϕ e θ. A precisão da amostragem dos ângulos de latitude e longitude devem crescer conforme se aumenta o valor de θ, dado que a distorção causada por uma latitude fixa ou uma longitude deslocada é mais drástica para altos valores de θ.

   Os parâmetros θ e ϕ são amostrados como mostrado na Figura 22 pelas seguintes regras:

   - As latitudes θ são amostradas, sendo que as inclinações associadas são dadas pela série geométrica $1, a, a^2, ..., a^n$, com $a > 1$. A escolha de $a = \sqrt{2}$ é um bom compromisso entre precisão e dispersão. Para a implementação desse método,



(a) Visão hemisférica de três valores de *tilt*

(b) Visão esférica superior com 41 pontos referentes ao intervalo de *tilt*

Figura 22 – Amostragem de parâmetros de latitude θ de longitude ϕ no ASIFT. As amostras são representadas pelos pontos de cor preta destacados nas imagens. (Adaptada de (YU; MOREL, 2011))

valores inteiros de n no intervalo $[5, 32[$ podem ser explorados.

- As longitudes ϕ são para cada inclinação uma série aritmética $0, b/t, ..., kb/t$, onde $b \simeq 72°$ apresenta-se como um bom compromisso entre, precisão e dispersão, e k é um inteiro tal que $kb/t < 180°$.

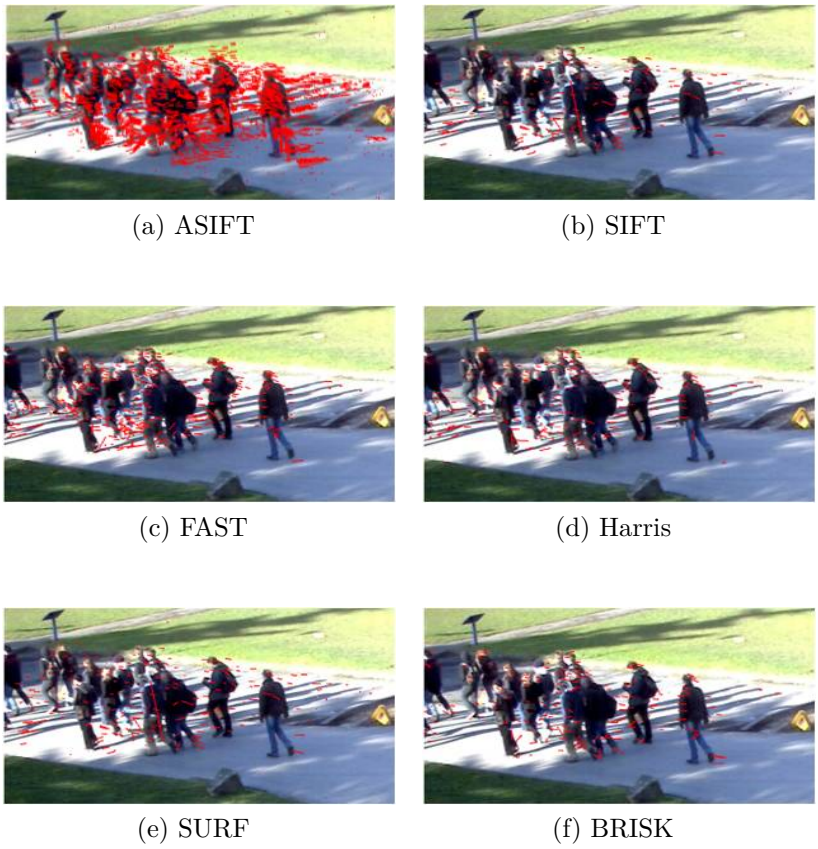
2. Todas as imagens simuladas são comparadas por um algoritmo de casamento invariante, neste caso SIFT (Seção 3.1.1), embora qualquer outro algoritmo pudesse ser implementado.

3. O método SIFT tem previsto em suas etapas critérios de eliminação de características irrelevantes (Seção 3.1.1.2). No entanto, usualmente SIFT traz consigo falsos positivos, mesmo para pares de imagens que não correspondem a mesma cena. O ASIFT, pela comparação de vários pares, pode entretanto acumular muitos erros, sendo importante um método para filtrá-los. Sendo assim, um método conhecido como ORSA (Algoritmo de Amostragem Aleatória Otimizada, do inglês, *Optimized Random Sampling Algorithm*) disponível por (MOISAN; STIVAL, 2004) é aplicado.

O ASIFT, por meio de simulações de variação do eixo de orientação da câmera, conhecidos como ângulos de latitude e longitude, detém dois novos parâmetros não previstos pelo SIFT. Dessa forma, o ASIFT consegue capturar muito mais informações relevantes do que o SIFT. A Figura 23 ilustra a extração de características em uma imagem usando ASIFT e os diferentes métodos avaliados em (FRADI; DUGELAY, 2015). Como pode ser observado, o ASIFT captura muito mais características do que todos os outros métodos

apresentados. Por este motivo, este trabalho propõe o uso do extrator ASIFT para a extração dos *keypoints*.

A saída do extrator neste trabalho é um vetor y_0 representando as coordenadas (x,y) de localização dos *keypoints* encontradas.



(a) ASIFT (b) SIFT

(c) FAST (d) Harris

(e) SURF (f) BRISK

Figura 23 – Extração de características por diferentes métodos.

3.2 Fluxo Óptico

O fluxo óptico pode ser compreendido como as velocidades aparentes de objetos em uma imagem por meio de um mapeamento vetorial conhecido como *Vetores de Deslocamento*. O fluxo óptico pode surgir tanto da movimentação dos objetos na imagem quanto do observador em relação a ela (GIBSON, 1950). Por consequência, o fluxo óptico pode oferecer importante informação sobre a distribuição espacial dos objetos visualizados e a taxa de deslocamento dessa distribuição.

A descontinuidade em um mapa de fluxo óptico pode auxiliar na segmentação de imagens, dado que essa descontinuidade pode corresponder a presença de diferentes objetos (SACHTLER; ZAIDI, 1995). As técnicas mais recentes que procuram resolver o fluxo óptico de imagens baseiam-se em duas vertentes: a conservação dos dados e a coerência

espacial. A coerência espacial surgiu para resolver de maneira eficiente a pressuposição feita para a primeira vertente, conservação do dado, de que as intensidades de pequenas regiões em imagens consecutivas permanecem constantes, mesmo que suas posições mudem. Essa pressuposição é regida pela Equação 3.7.

$$I(x, y, t) = I(x + u\delta t, y + v\delta t, \delta t), \tag{3.7}$$

onde $I(x, y, t)$ é a representação de uma imagem com intensidades na escala de cinza, $\mathbf{d} = (u, v)^T$ é o deslocamento de um ponto e δt é um pequeno intervalo de tempo para uma posição $\mathbf{x} = (x, y)$.

Para solução da Equação 3.7, deve-se encontrar $\mathbf{d}$ o qual recai em um sistema linear não determinado após aplicação da aproximação de Taylor de primeira ordem. As técnicas de coerência espacial foram introduzidas (SENST; EISELEIN; SIKORA, 2010; HORN; SCHUNCK, 1981), sendo categorizadas como uma restrição local que sustenta a afirmativa de que o deslocamento de uma pequena região para *frames* consecutivos é constante.

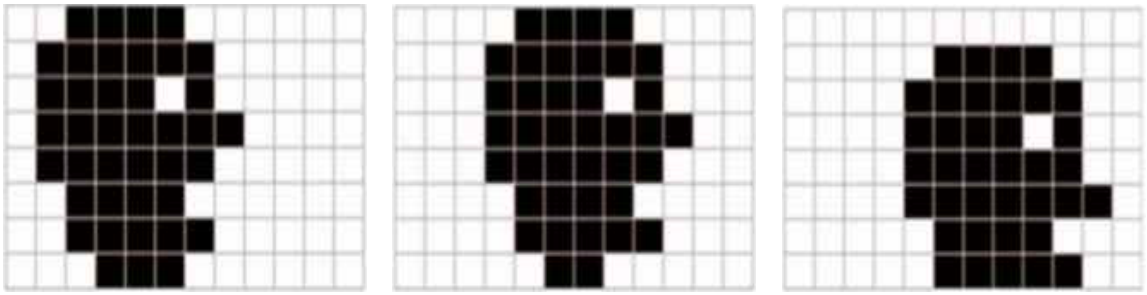
Embora uma série de trabalhos tragam estados da arte em relação ao cálculo de fluxo óptico de imagens, grande parte desses ainda utilizam a popular técnica proposta por (LUCAS; KANADE, 1981), sendo que os trabalhos subsequentes apostaram na melhora de seu desempenho para aplicações em tempo real (GARRIGUES; MANZANERA, 2012; ZACH; GALLUP; FRAHM, 2008; SINHA et al., 2006).

A Figura 24 representa um mapeamento de vetores de deslocamento para uma uma sequência de *frames* consecutivos assumindo a conservação da intensidade.

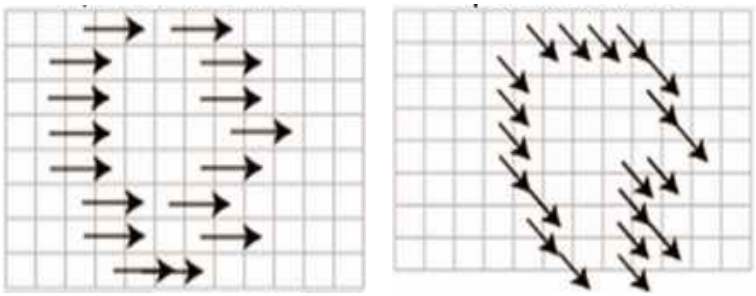
3.2.1 Fluxo Óptico Local Robusto Baseado em Cruz

Neste trabalho propõe-se o uso do fluxo óptico CBRLOF (SENST et al., 2014), um método mais preciso e de melhor desempenho computacional do que outros métodos clássicos.

O método de fluxo óptico CBRLOF (Fluxo Óptico Local Baseado em Cruz, do inglês, *Cross-Based Robust Local Optical Flow*) proposto em (SENST et al., 2014) pode ser considerado uma extensão de outras duas metodologias de fluxo óptico local, ainda muito utilizadas em aplicações de computação visual: a PLK (Lucas Kanade Piramidal, do inglês *Pyramidal Lucas Kanade*) (BOUGUET, 2000) e RLOF (Fluxo Óptico Local Robusto, do inglês, *Robust Local Optical Flow*) (SENST; EISELEIN; SIKORA, 2012). Essa extensão faz uso da geração de variáveis de suporte baseadas em cruz, propostas por (ZHANG; LU; LAFRUIT, 2009), que procuram pelas descontinuidades das intensidades locais das imagens. A geração dessas variáveis usam uma "região de suporte" adaptativa que respeita os limites da imagem. Tanto a RLOF e PLK diferem-se dessa técnica baseada em cruz por



(a) *frame 1* (b) *frame 2* (c) *frame 3*



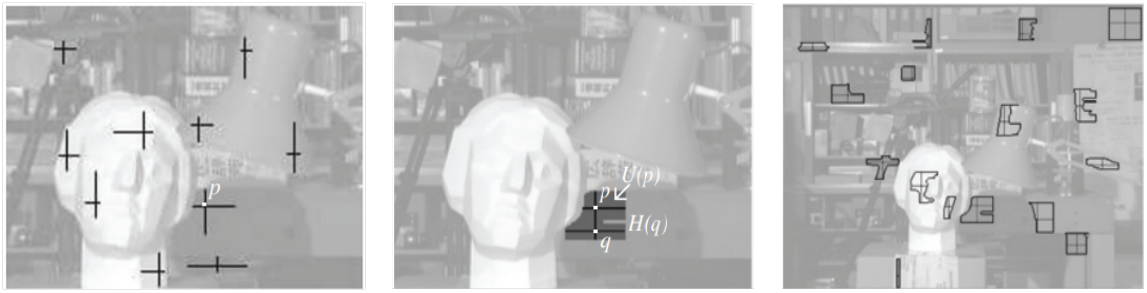
(d) Fluxo Óptico *frame* 1-2 (e) Fluxo Óptico *frame* 2-3

Figura 24 – Deteção de fluxo óptico em 3 imagens temporariamente consecutivas. (Adaptado de (MILITELLO; RUNDO; GILARDI, 2014)).

possuírem a limitação de generalizar essa "região de suporte" em um formato retangular, não se adaptando aos reais contornos dos objetos da imagem. Neste caso, uma região retangular tem maior probabilidade de conter o movimento de mais de um objeto, o que é indesejado.

A ideia principal da construção de "regiões de suporte" $U(p)$ baseada em cruz é decidir uma cruz vertical para cada *pixel* $p = (x_p, y_p)$ baseado na informação de similaridade de cor em uma imagem I. Essa similaridade de cor é encontrada a partir de um parâmetro τ atribuído empiracamente, trabalhando-se no espação RGB como mostrado em (ZHANG; LU; LAFRUIT, 2009). Como mostrado na Figura 25a, essa cruz adaptativa consiste de duas linhas ortogonais que se interceptam em um *pixel* p (*pixel* âncora). Os segmentos horizontais são representados por $H(p)$ e os verticais por $V(p)$. Assim, valores de $h_p^-, h_p^+, v_p^-, v_p^+$ que representam os tamanhos dos braços dos seguimentos é o que se procura para definir as regiões de suporte para auxílio no cálculo do fluxo óptico. A Figura 25c mostra regiões de suporte em uma imagem I, e um detalhamento de uma região de suporte pode ser vista na Figura 26 com as representações das variáveis mencionadas acima.

A "região de suporte" $U(p)$ (Equação 3.2.1) pode ser equacionada como uma área integrada de múltiplos segmentos horizontais $H(p)$ que se apóia sobre um segmento vertical



(a) Cruzes Verticais mapeadas para *pixels* em uma imagem. I

(b) Região de Suporte $U(p)$ dinamicamente construída para um *pixel* p, integrando múltiplos seguimentos $H(q)$ das cruzes vizinhas.

(c) Regiões de suporte aproximado dos contornos/estruturas de objetos em uma imagem.

Figura 25 – Representação e construção de Regiões de Suporte baseadas em cruz. (Adaptado de (ZHANG; LU; LAFRUIT, 2009)).

$V(p)$ para o *pixel* p.

$$U(p) = \bigcup^{q \epsilon V(p)} H(p) \tag{3.8}$$

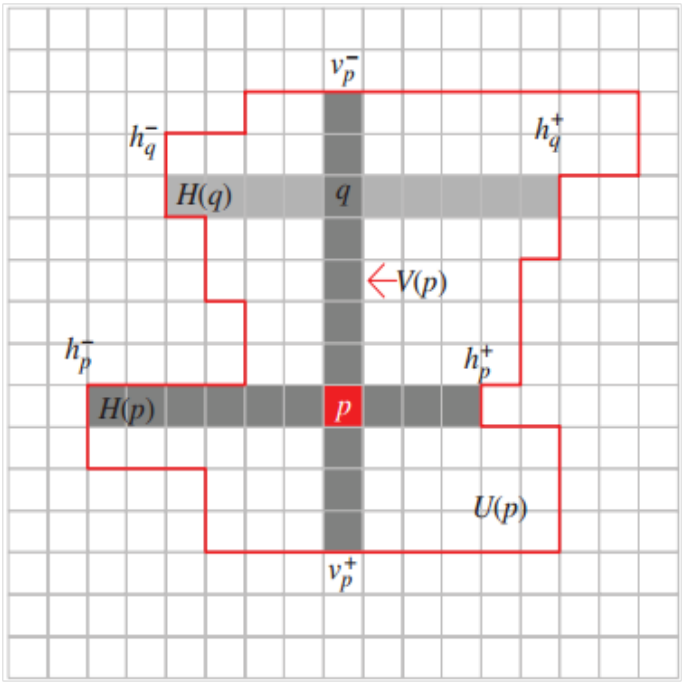


Figura 26 – Configuração de uma cruz vertical $H(p) \cup V(p)$ para um *pixel* âncora p, e a região de suporte adaptada $U(p)$. $q \epsilon V(p)$ é um *pixel* no segmento vertical V(p). (Adaptado de (ZHANG; LU; LAFRUIT, 2009)).

A restrição definida por Lucas Kanade de movimento constante assume um único componente de movimento em uma região de suporte. Assim, para cada região de suporte

$U(p)$ define-se um vetor deslocamento $\mathbf{d}$, Equação 3.7, para um *pixel* âncora p. Em (SENST et al., 2014) é comprovada a boa desempenho dos resultados e desempenho da técnica perante abordagens anterioes, mostrando que a técnica estima um maior número de vetores de movimentos ao redor dos objetos.

3.3 Projeção *Forward-Backward*

Estimula-se neste trabalho o uso de uma filtragem, proposta por (FRADI; DUGELAY, 2015), específica para os *keypoints* encontrados pela ASIFT em conjunto com a técnica de fluxo óptico CBRLOF. Esta filtragem envolve a remoção de *keypoints* de baixa relevância para a representação da localização de pessoas, o que traz uma maior robustez sobre as características resultantes. Os passos seguintes descrevem essa filtragem.

Para cada coordenada de *pixel* de um vetor y_0 do *frame* I_k, contendo *keypoints* extraídos da ASIFT, verifica-se, pelo fluxo óptico, a sua projeção sobre o *frame* seguinte I_{k+1}, gerando o vetor y_1 de coordenadas (x,y) com as projeções encontradas (passo de *forward*). De maneira similar, faz-se o caminho inverso, verificando as projeções do conteúdo de y_1, do *frame* I_{k+1} sobre I_k gerando o vetor y_2 (passo de *backward*).

As partes superior e inferior da Figura 27 representam, na sequência, as etapas de *forward* (y_1 a partir de y_0) e *backward* (y_2 a partir de y_1).

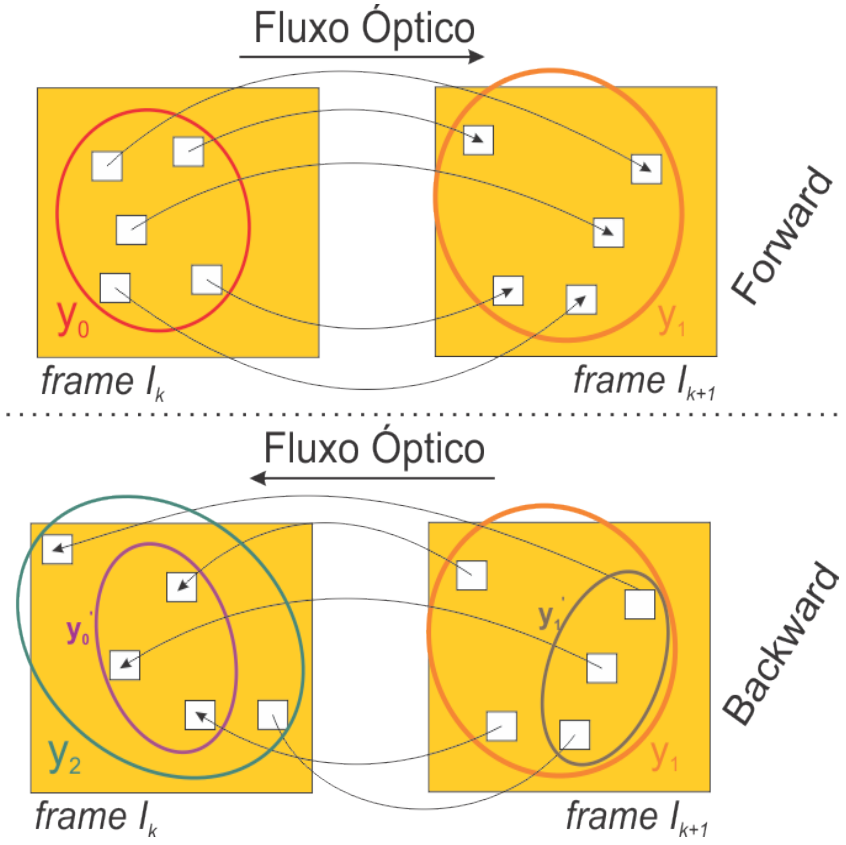


Figura 27 – Processo *Forward-Backward*.

As coordenadas $y_0^{'}$ resultante da operação $y_0^{'}=y_0 \bigcap y_2$ representam os *pixels* de saída deste processo conhecido como *forward-backward*, sendo que o conteúdo do vetor $y_1^{'}$ representa os valores das coordenadas de y_1 que foram projetadas em $y_0^{'}$.

Mesmo com o filtro de *keypoints* realizado pelo roteiro descrito acima, ainda não é suficiente para garantir que outras características irrelevantes sejam descartadas. Para isso, adiciona-se ainda um passo de descarte de informações de pequenas trajetórias. Essas pequenas trajetórias estão normalmente associadas a pequenos ruídos, imperfeições do sistema de aquisição ou movimentos irrelevantes existentes em uma sequência de *frames*. Essa etapa de eliminação de pequenas trajetórias segue o seguinte:

1. A partir das coordenadas de *pixel* em $y_0^{'}$ e $y_1^{'}$ calcula-se as distâncias euclidianas dos pixels relacionados.
2. Coordenadas com distâncias $d^{'}$ menores que um dado limiar ζ são eliminadas.
3. As coordenadas restantes são representadas pelo vetor $y_0^{''}$ que correspondem às coordenadas de *pixels* finais a serem utilizadas em uma construção de mapa de densidade.

3.4 Construção do Mapa de Densidade

A etapa de cálculo dos mapas de densidade assume que as características remanescentes do processo de *forward-backward* da Seção 3.3 possuem igual contribuição na imagem, ou seja, regiões com maior concentração de características representam um maior movimento (alta densidade) de pessoas e regiões com menor concentração representam menor movimento (baixa densidade) de pessoas em um *frame* I_k.

O mapa de densidade C_k é estimado por uma soma de funções de densidade de probabilidade (pdf) Gaussianas representada na Equação 3.9 a partir das características locais presentes no vetor $y_{0k}^{''}$, resultante da Seção 3.3. Essa soma permite inferir a contribuição de cada pixel (x,y) do *frame* I_k para o mapa total de densidade da imagem.

$$C_k(x,y) = \frac{1}{\sqrt{2\pi\sigma}} \sum_{i=1}^{y_{0k}^{''}} e^{-(\frac{(x-x_i)^2+(y-y_i)^2)}{2\sigma^2})}, \tag{3.9}$$

onde σ é a largura de banda definida para uma sequência de imagens. A matriz $C_k(x,y)$ é então normalizada para valores entre 0 e 1. Um resultado visual da Equação 3.9 pode ser visto na Figura 17b.

O fluxograma da Figura 28 mostra de forma resumida a interação dos processos descritos até aqui para o desenvolvimento do mapa de densidade.

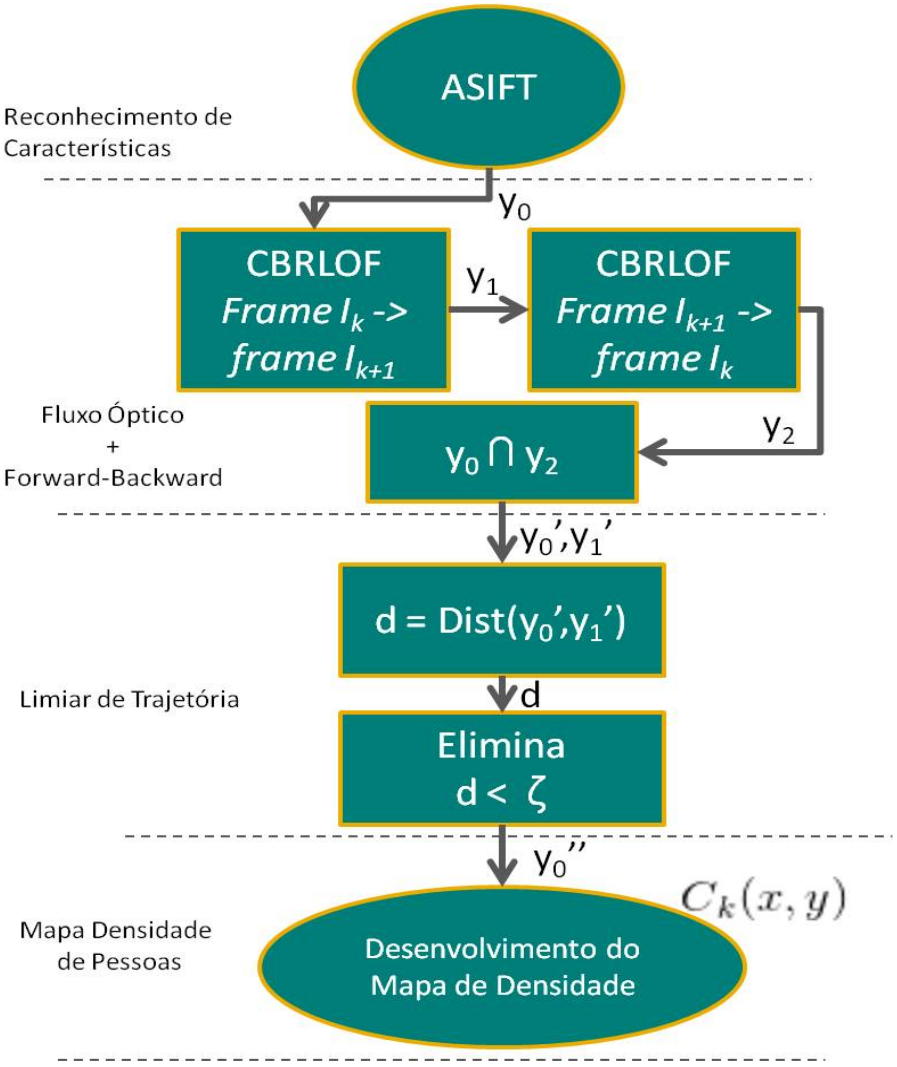


Figura 28 – Procedimento de Cálculo do Mapa de Densidade para uma sequência de *frames*.

4 Integração de Detector de Pedestre com o Mapa de Densidade e Outros Métodos Propostos

Além do uso da técnica ASIFT para extração de características locais das imagens (Subseção 3.1.2) e da CBRLOF para o cálculo de fluxo óptico (Subseção 3.2.1), ambas utilizadas na construção do mapa de densidade (Seção 3.4) *frame* a *frame* em uma sequência de imagens, a metodologia proposta neste trabalho baseia-se na integração a ser realizada entre as *boxes* do detector de pedestres e os mapas de densidade.

A integração entre os resultados do detector e mapa de densidade é viabilizada partindo-se de filtragens que levam em consideração as características dimensionais e localização das *boxes* de saída do detector de pedestres (Subseção 4.1.1), seguido por um processo de filtragem realizado com a informação dos mapas de densidade construídos (Subseção 4.1.2), e que irão oferecer uma maior precisão ao resultado final das detecções por redução de falsos positivos.

Existem ainda modificações adicionais propostas e implementadas como produto deste trabalho que, além do que já existe na literatura sobre este processo de integração, visam melhorar a acurácia e a robustez da detecção de pedestres em grupos de pessoas. Estas modificações extras são representadas por: 1) etapas de predição de hipóteses em um *frame* I_k a partir das *boxes* detectadas no *frame* I_{k-1} (Subseção 4.3) e 2) a integração da detecção de detectores distintos (Subseção 4.4). A viabilidade do tema abordado de integração de detecção de pedestres por técnicas distintas da Subseção 4.4 só é possível devido à normalização dos *scores* de técnicas de detecção de pedestres (Subseção 4.2), e o cálculo adaptativo de limiares máximo e mínimo para construção dos mapas de densidade, conforme apresentado na Subseção 4.2.

4.1 Integração Mapa de Densidades e Detector

A partir das informações sobre o mapa de densidades gerado na Seção 3, (FRADI; DUGELAY, 2015) propõem-se uma forma de otimização do comportamento dos detectores de pessoas pela aplicação de filtros e utilização de um limiar dinâmico (τ_{din}) calculado para cada uma das detecções. Um dos maiores benefícios do emprego desta rotina está relacionado com a redução de falsos positivos deixando o detector mais preciso.

O processo de integração compreende uma etapa inicial de restrições geométricas

que relacionam as razões de largura e altura das detecções e predições de detecções entre o *frame* I_{k-1} e I_k. Uma segunda etapa, chamada de NMS (*Non-Maximum Supression*), verifica a sobreposição das detecções e a sua relevância quanto à densidade daquela região.

4.1.1 Restrições Geométricas

A aplicação das restrições geométricas neste ponto é sugerida devido às características do modelo de partes deformáveis, que pode mesclar partes de pessoas diferentes em uma única detecção. Essas restrições são aplicadas em duas etapas. Seja D_k o conjunto de detecções realizadas no *frame* I_k, a Equação 4.1, referente à primeira etapa, relaciona as alturas h_j^k e os pontos y_j^k das detecções de I_k, permitindo por regressão linear que os valores de $\alpha_{\text{k-1}}$ e $\beta_{\text{k-1}}$ sejam encontrados:

$$h_j^k = \alpha_{k-1} \cdot y_j^k + \beta_{k-1} + \epsilon, \qquad j\epsilon \{1...n_k\}. \tag{4.1}$$

Apenas detecções que atenderem à restrição de erro de altura dada por $\left|h_j^k - \tilde{h}_j^k\right| \leq \Delta_h$ são aceitas, onde $\tilde{h}_j^k$ são as alturas projetadas a partir da Equação 4.1 pelos dados de y_j^k, e Δ_h é um certo limiar de aceite ajustado pelo usuário. O erro ϵ é ocasionado pela imprecisão do modelo.

Uma segunda restrição geométrica é dada pelas razões de largura e altura das detecções. A Equação 4.2 introduz o parâmetro γ_{k-1} que permite a visualização dessa relação:

$$\gamma_{k-1} = mediana \left\{ \frac{w_j^i}{h_j^i} \right\}_{1 \leq i \leq (k-1), 1 \leq j \leq n_i}. \tag{4.2}$$

Observe que na Equação 4.2 os valores de i variam desde o primeiro *frame* da sequência até o penúltimo *frame* $I_{k\text{-}1}$, ou seja, n_i representa o número de detecções acumuladas até o *frame* i. O valor de γ_{k-1} encontrado é então comparado na inequação de erro $\left|\left(\frac{w_j^i}{h_j^i}\right) - \gamma_{k-1}\right| \leq \Delta_\gamma$, sendo que detecções em que a razão $\left(\frac{w_j^i}{h_j^i}\right)$ não atenderem à inequação são descartadas. Δ_γ é um escalar de aceite ajustado pelo usuário. Os parâmetros α_k, β_k e γ_k são atualizados a cada nova rodada do cálculo do mapa de densidade.

4.1.2 Restrições NMS

As restrições NMS permitem a eliminação de detecções sobrepostas D_k e cálculo dos limiares dinâmicos (τ_{din}) para cada detecção de D_k. A etapa inicial testa as superposições de todas as detecções d_j^k de um *frame* I_k entre si, sendo que na identificação de sobreposições que ultrapassem um dado valor Δ_0, atribuída pelo usuário, as *boxes* de menores *scores* são eliminadas.

Uma segunda etapa de cálculo de limiar de detecção dinâmica (τ_{din}) para cada detecção d_j^k é dada pelas Equações 3.9, 4.3 e 4.4.

$$\hat{C}_k(d_j^k) = \frac{\sum_{p=0}^{h_j^k-1}\sum_{q=0}^{w_j^k-1} C_k(x_j^k+p, y_j^k+q)}{h_j^k \cdot w_j^k}, \tag{4.3}$$

$$\tau_{din} = \tau_{max} + (\tau_{min} - \tau_{max}) \cdot \hat{C}_k(d_j^k), j\epsilon\{1, ..., n_k\} \ , \tag{4.4}$$

onde $\tau_{max} = -0.5$ e $\tau_{min} = -1.2$ foram empiracamente encontrados e fornecidos em (FRADI; DUGELAY, 2015).

Em resumo, valores de τ_{din} são calculados para cada detecção pela equação 4.4 que envolve a soma de todas as densidades mapeadas para cada *pixel* dentro dos limites das *boxes* de hipóteses encontradas pelo detector. Hipóteses que tenham um *scores* menor do que τ_{din} são eliminadas, por serem consideradas fora da região de maior probabilidade de existência de pessoas.

Um fato importante a se considerar é que o mapa de densidade aqui desenvolvido trabalha com as características que se movimentam nas imagens, neste caso movimentação de pessoas. Sendo assim, é razoável entender que o caso de pessoas paradas entre dois *frames* não gera movimento, não gerando fluxo óptico relevante naquela região. Essa falta de movimento é calculado como uma região de baixa densidade de pessoas. Dessa forma, é possível que algum "verdadeiro positivo" seja removido pela integração entre saída de detector de pedestres para um *frame* I com um mapa de densidade C.

4.2 Normalização dos *scores* e Cálculo de Limiares Adaptativos

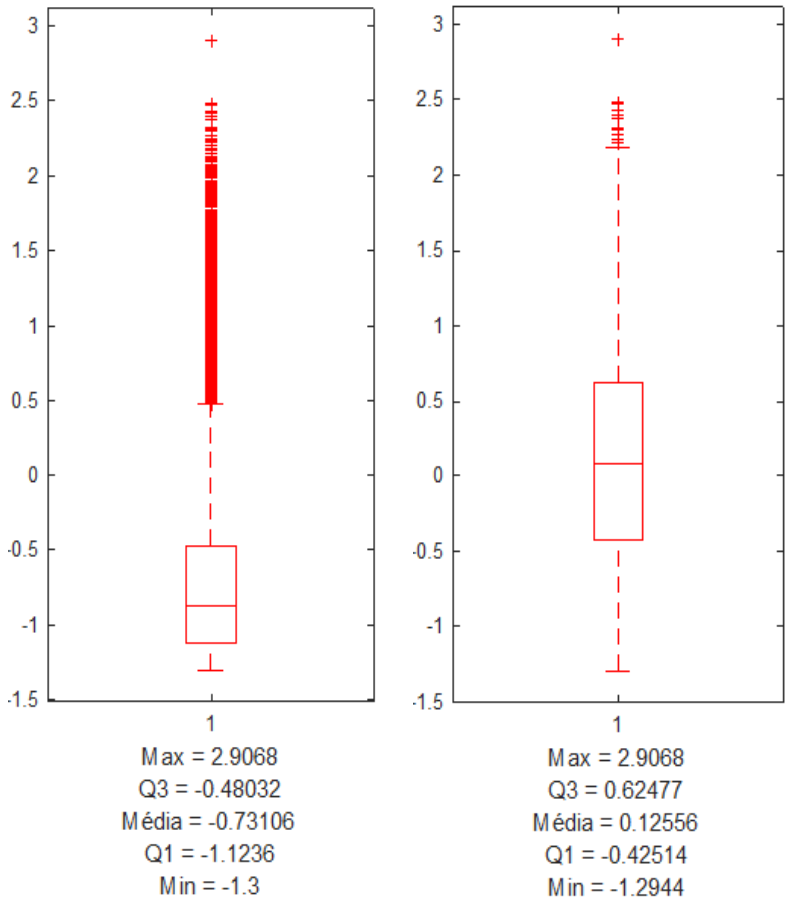
Ao utilizar diferentes técnicas para detecção de pessoas, visualiza-se que os resultados das hipóteses detectadas são distintos, seja em relação à precisão, à magnitude dos *scores*, à quantidade de falsos positivos ou perdas ("misses") de objetos.

Uma primeira proposta deste trabalho para reduzir essa diferença está sobre a compatibilização dos *scores*. A normalização dos *scores* apresentada neste trabalho é uma forma de compatibilizar a leitura dos *scores* de diferentes técnicas de detecção de uma forma única. Isso implica em normalizar os *scores* para uma nova faixa de valores dentro do conjunto [0,1]. Essa normalização permite uma comparação mínima de diferentes técnicas de detecção de pedestres. Esse passo de normalização dos *scores* é importante ainda para a adaptabilidade proposta a seguir, de forma a calcular limiares adaptativos para detectores distintos por um mesmo algoritmo.

Essa normalização proporciona explorar uma forma de gerar limiares adaptativos para os valores de τ_{max} e τ_{min} introduzidos na Subseção 4.1.1 sem a necessidade de atribuição

de um valor fixo (τ "Não Adaptativo"), chamado neste trabalho de τ "Adaptativo". Para este estudo foram usados como referência os valores de $\tau_{max} = -0,5$ e $\tau_{min} = -1,2$ sugeridos empiricamente em (FRADI; DUGELAY, 2015).

A abordagem realizada para aferição dos valores de $\tau_{max} = -0,5$ e $\tau_{min} = -1,2$ não é descrita em (FRADI; DUGELAY, 2015). Dessa forma, a fim de experimentar uma abordagem para encontrar valores compatíveis de τ_{max} e τ_{min}, uma primeira análise deste trabalho verificou o comportamento dos *scores* do detector MDPM (Seção 2.3) utilizado em (FRADI; DUGELAY, 2015) para a base de dados PETS2009-S1L1-1-(13-57) (Seção 5.1) quando parametrizado a fornecer muitas hipóteses "falsas positivas". O resultado foi analisado por meio da ferramenta de análise descritiva *boxplot* e pode ser visualizado na Figura 29a.



(a) *BoxPlot* com *scores* do detector parametrizado para fornecer falsos positivos.

(b) *BoxPlot* com *scores* das hipóteses verdadeiras após comparação das hipóteses Figura 29a com o *ground-truth*.

Figura 29 – *BoxPlots* construídos a partir dos *scores* do detector MDPM para a base de dados PETS2009-S1L1-1-(13-57).

Em um segundo momento comparou-se as hipóteses que geraram o gráfico da Figura 29a, com o *ground-truth* da base de dados. O resultado encontrado pode ser visualizado na Figura 29b. O mesmo teste foi realizado com outras bases de dados do Banco PETS2009 e foi verificado comportamento similar entre todas elas.

Comparando os valores de $\tau_{max} = -0,5$ e $\tau_{min} = -1,2$ sugeridos empiricamente em (FRADI; DUGELAY, 2015), com o *BoxPlot* da Figura 29b, percebe-se que o valor de $\tau_{max} = -0,5$ é muito próximo ao valor do primeiro quartil ($Q1 = -0,42514$), e o valor de $\tau_{min} = -1,2$ é muito próximo do valor mínimo ($Min = -1,2944$). Essa observação para atribuição do valor de τ_{max} e τ_{min} será aceita considerando que dentro dessa região de *scores* o detector informa hipóteses duvidosas quanto à possibilidade de ser um pedestre válido.

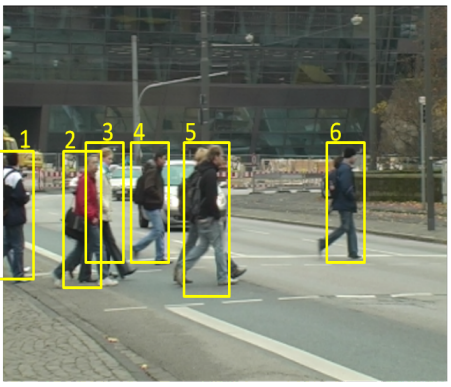
A partir da observação relatada acima, foi estipulado que ao passo de cada *frame* os valores de τ_{max} e τ_{min} serão calculados da seguinte forma:

- O valor $\tau_{max(k)}$ para um *frame* I_k será igual ao primeiro quartil (Q1) dos *scores* das hipóteses encontradas pelo detector para um *frame* I_k.
- O valor $\tau_{min(k)}$ para um *frame* I_k será igual ao valor mínimo dos *scores* das hipóteses encontradas pelo detector atualizado a cada *frame*. Ao passo de que os dados foram normalizados entre [0,1], o valor de $\tau_{min(k)}$ será 0.

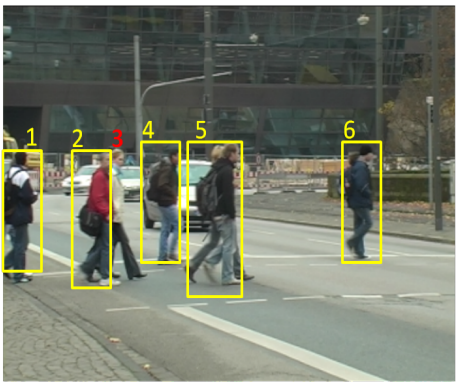
4.3 Predição de Hipótese de Detecção de Pedestres em *frames* consecutivos

Com o intuito de observar a colaboração das hipóteses de detecções de um *frame* I_{k-1} para o *frame* I_k seguinte, foi desenvolvida uma abordagem onde se utiliza as características restantes após a etapa *forwared-backward* (Seção 3.3) para prever onde as hipóteses D_{k-1} estariam no *frame* I_k, visando suprir a perda de alguma hipótese não encontrada pelo detector no conjunto de hipóteses D_k. Neste sentido, confia-se na acurácia das hipóteses D_{k-1} que já passaram pelos filtros após os mapas de densidade.

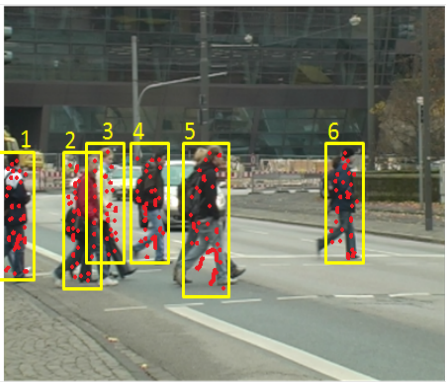
A Figura 30 ilustra o caso citado acima. Percebe-se na Figura 30a uma sequência de seis hipóteses D_{k-1} de pessoas mapeadas por um detector. O mesmo detector aplicado a uma *frame* I_k apresenta a saída D_k apresentada na Figura 30b. Perceba que as hipóteses em D_k não contemplam o indivíduo rotulado com o número três.



(a) Hipóteses de Localização de pessoas D_{k-1} para um *frame* I_{k-1} de uma sequência I.

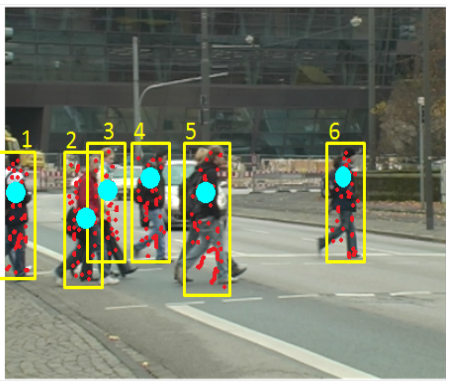


(b) Hipóteses de Localização de pessoas D_k para um *frame* I_k de uma sequência I.

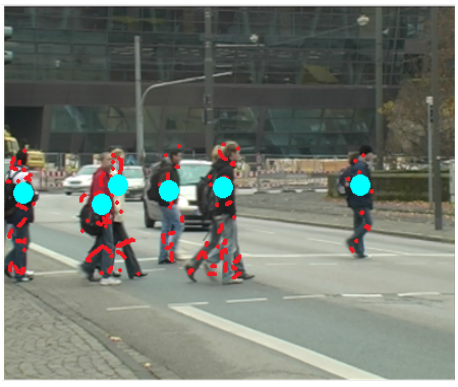


(c) Características localizadas nos limites das hipóteses D_{k-1} para um *frame* I_{k-1} de uma sequência I.

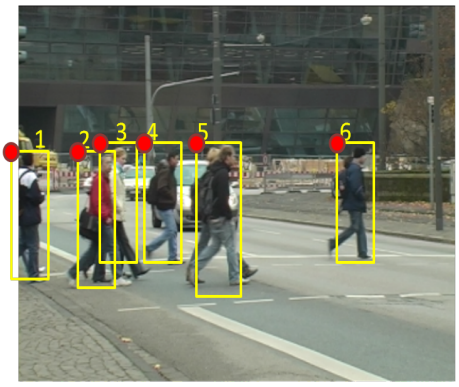
(d) Características localizadas nos limites das hipóteses D_k para um *frame* I_k de uma sequência I.



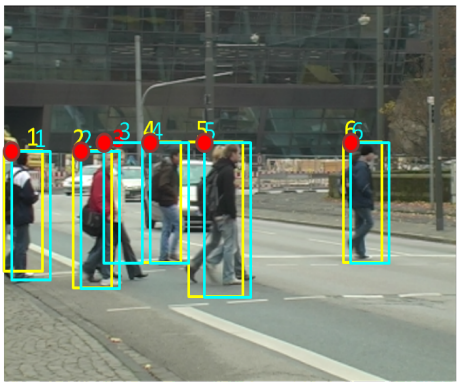
(e) controides calculados a partir das características localizadas nos limites das hipóteses D_{k-1} para um *frame* I_{k-1} de uma sequência I.



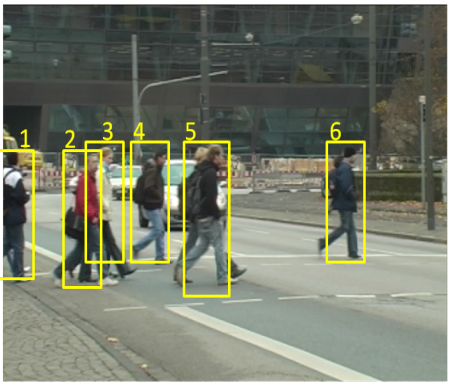
(f) controides calculados a partir das características localizadas nos limites das hipóteses D_k para um *frame* I_k de uma sequência I.



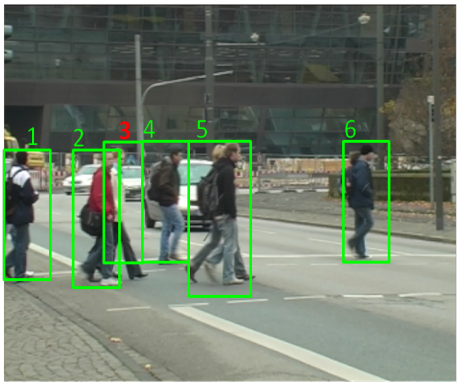
(g) Hipóteses de Localização de pessoas D_{k-1} para um *frame* I_{k-1} de uma sequência I.



(h) Hipóteses de Localização de pessoas D_k para um *frame* I_k de uma sequência I.



(i) Hipóteses de Localização de pessoas D_{k-1} para um *frame* I_{k-1} de uma sequência I.



(j) Hipóteses de Localização de pessoas D_k para um *frame* I_k de uma sequência I.

Figura 30 – Predição de hipóteses de pessoas em *frames* consecutivos.

A fim de mitigar situações como a descrita acima, os seguintes passos são realizados para uma sequência de dois *frames* consecutivos:

1. Os *pixels* de características locais do frame I_{k-1} (vetor $y_0^{'}$ da Seção 3.3) localizados dentro da fronteira limitada pela *box* de uma hipótese d_j^{k-1} são selecionados (Figura 30c).

2. Os *pixels* de características locais do *frame* I_k (vetor $y_1^{'}$ da Seção 3.3) localizados dentro da fronteira limitada pela *box* de uma hipótese d_j^k são selecionados (Figura 30d).

3. A partir dos *pixels* de I_{k-1} selecionados, calcula-se a coordenada dos centróides (Cen_j^{k-1}) que representam esses grupos de *pixels* (Figura 30e).

4. A partir dos *pixels* de I_k selecionados, calcula-se a coordenada dos centróides (Cen_j^k) que representam esses grupos de *pixels* (Figura 30f).

5. Para cada centroide Cen_j^{k-1} e Cen_j^k, calcula-se a distância entre suas coordenadas.

6. As distâncias encontradas entre os centroides são utilizadas para "ajustar" a nova posição do canto superior esquerdo das *boxes* do *frame* I_{k-1}, para prever onde estariam posicionadas no *frame* I_k. A Figura 30h apresenta estas *boxes* na cor ciano com hipóteses D_k em amarelo.

A Figura 30h apresenta o resultado do processo até aqui, onde se tem as *boxes*, na cor amarela, de hipóteses do detector para o *frame* I_k (D_k) e as *boxes*, na cor ciano, que representam as hipóteses de pessoas H_k. O ganho esperado é poder atribuir a hipótese 3 de H_k às hipóteses de *frame* I_k.

Com o intuito de interagir as *boxes* de D_k e H_k evitando redundâncias de hipóteses para uma mesma pessoa, utiliza-se uma etapa chamada aqui de Clusterização Hierárquica Binária. Nesta etapa, todas as localizações e dimensões das hipóteses em D_k e H_k da Figura 30h têm suas similaridades analisadas. Essa similaridade é verificada visualmente no dendrograma da Figura 31 onde, a partir da Clusterização Hierárquica das localizações e dimensões das *boxes*, agrupa-se aquelas do ponto mais inferior do dendrograma duas à duas. Do exemplo do gráfico da Figura 31, observa-se o subconjunto $\{\{2,10\},\{5,8\},\{1,4\},\{3,6\},\{7,11\}\}$, representando hipóteses similares e o subconjunto $\{9\}$ representado uma hipótese sem correspondência direta. Este conjunto $\{9\}$, no exemplo da Figura 30, representa a hipótese 3 existente em D_{k-1} e que não existe em D_k.

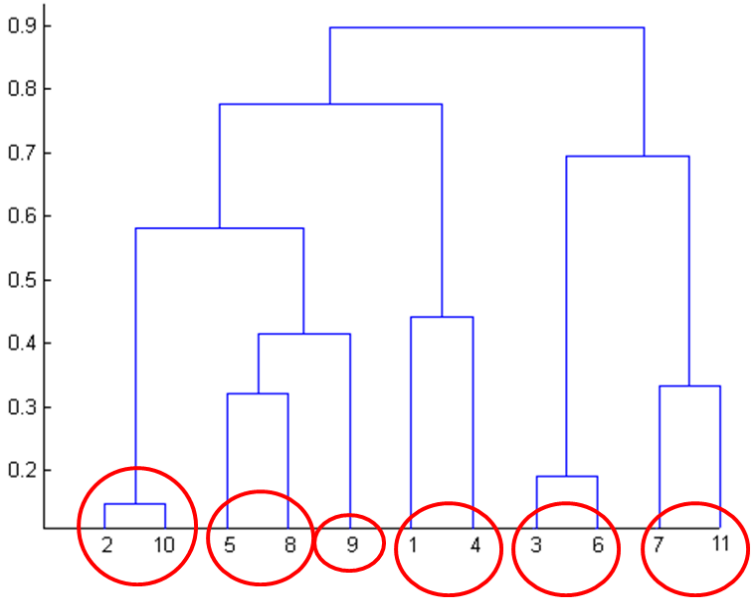


Figura 31 – Dendrograma de similaridade entre as hipóteses de D_k e H_k.

A partir da similaridade visualizada no dendrograma da Figura 31, as hipóteses com alguma correspondência direta são "mescladas", duas a duas, tomando-se a média das dimensões de largura e altura, e atribuindo o *score* mais alto à hipótese final. O resultado final deste processo pode ser visualizado na Figura 30j. As hipóteses isoladas sem correspondências são simplesmente adicionadas ao novo conjunto de hipóteses para o *frame* I_k.

4.4 Integração entre Hipóteses de Detectores de Pedestres Distintos

A integração entre as hipóteses de saída de técnicas de detecção distintas, sugerida nesta seção, trata-se de uma abordagem que se aproveita da melhora esperada nos resultados obtidos com a aplicação das técnicas das Seções 4.1 à 4.3. Sendo assim, até este ponto, entende-se que existem ganhos quanto à redução de "falsos positivos" e de forma mais "tímida" a redução de perdas pela aplicação dessas técnicas sobre às hipóteses de saída de detectores de pedestres.

Para um *frame* I_k, considera-se o conjunto D_k^A como sendo o conjunto de hipóteses de pedestres de um *Detector* A após a aplicação das técnicas anteriores. De forma análoga, o conjunto D_k^B representa as hipóteses do *Detector* B para o *frame* I_k. De forma similar ao que foi exposto na Seção 4.3, a fim de evitar hipóteses redundantes para um mesmo pedestre, utiliza-se clusterização hierárquica binária para "mesclar" detecções similares. O resultado desta etapa dé o conjunto formado por todas as hipóteses de pedestres resultantes dessa união, ou seja, a união é realizada pela clusterização hierárquica binária. A saída desta abordagem deve aumentar o número de "falsos positivos" de saída, mas a redução de "perdas" deve compensar esse aumento que no final trará um ganho quanto ao cálculo das métricas a serem detalhadas na Seção 5.2. A Figura 32 representa de forma visual a união proposta como resultado desta etapa.

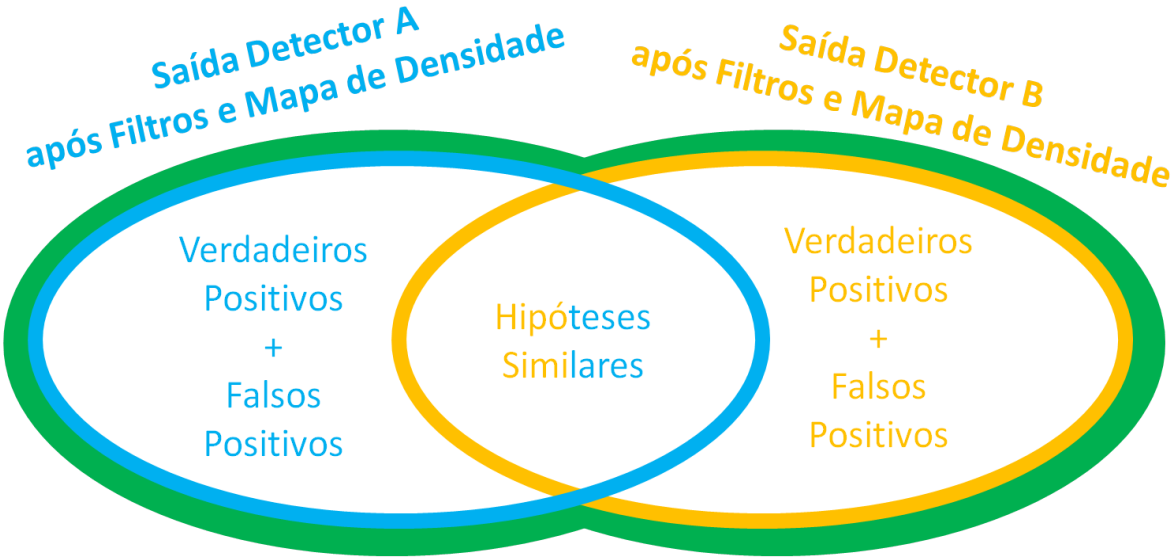


Figura 32 – Representação da integração entre detectores proposta neste trabalho.

5 Experimentos

Neste capítulo são apresentadas respostas observadas com a utilização da abordagem proposta, quando aplicada sobre bases de dados de vídeos contendo grupos de pessoas. Para isso, este capítulo traz na Seção 5.1 uma visão do cenário encontrado em cada sequência de imagens. Em seguida, a Seção 5.2 traz conceitos preliminares e indicadores a serem calculados para viabilizar a avaliação das respostas. Na Seção 5.3 os parâmetros configuráveis utilizados neste trabalho, tanto para os detectores quanto para o mapa de densidade, são expostos. A seção seguinte (Seção 5.4) concentra as apresentações das respostas em diferentes visões, como: 1) a precisão da utilização da técnica de extração de características ASIFT (Subseção 5.4.1), 2) o resultado da metodologia aplicada em detectores individuais e 3) as saídas encontradas após a etapa de integração das respostas dos detectores MDPM e LDCF filtradas pelas restrições e os mapas de densidade (Subseção 5.4.3).

5.1 Base de Dados de Imagens

As base de dados de vídeos utilizadas neste trabalho foram adquiridas de fontes distintas e possuem capacidades específicas de informações, pois variam em características como: quantidade de pedestres por *frame*, concentração de pessoas por área do *frame*, oclusões de pedestres, distância dos pedestres em relação à câmera, iluminação ambiente e *background*. As subseções a seguir resumem pontos relevantes das bases de dados trabalhadas.

5.1.1 INRIA

Para o treinamento dos detectores de pedestres foi utilizado o banco de dados de imagens de pessoas INRIA[3], produzida em (DALAL; TRIGGS, 2005). Esta base de dados possui uma coleção de imagens estáticas em compressão jpeg, de dimensões variadas, que contém pessoas em poses, aparências, vestimentas, iluminação e *background* variadas. Os indivíduos estão sempre em pé e em alguns casos existem oclusões de partes. A Figura 33 mostra exemplos de imagens integrantes da base de dados INRIA, utilizadas neste trabalho, e que são utilizadas como imagens positivas (614 imagens com 1237 anotações) que contêm anotações de treinamento, e 1218 imagens negativas, onde não existem pessoas e portanto sem anotações.

[3] *http://lear.inrialpes.fr/data*

(a) Exemplos Positivos.

(b) Exemplos Negativos.

Figura 33 – Exemplo de imagens da base de dados INRIA utilizadas no treinamento dos detectores de pedestres deste trabalho.

5.1.2 PETS2009

A base de dados PETS2009[4] apresenta cenas gravadas na *University of Reading, UK* com cenários de complexidades distintas. A PETS2009 é composta por imagens de 768×576 *pixels*, em compressão jpeg, que se dividem nas sequências *S1*, *S2* e *S3*, sendo que cada sequência fornece uma característica diferente de distribuição de pedestres. A Tabela 2 resume as principais características de cada sequência da base de dados PETS2009 utilizada neste trabalho e divulgadas em (GE; COLLINS, 2009).

A Figura 34 mostra exemplos de *frames* extraídos das sequências mencionadas na Tabela 2.

5.1.3 TUD

As bases de dados TUD-Campus, TUD-Crossing e TUD-Stadtmitte possuem similaridade quando à dispersão das pessoas, pois tratam-se de imagens de 640×480 *pixels*

[4] *http://cvg.reading.ac.uk/PETS2009*

Tabela 2 – Principais características da Base de Dados PETS2009.

sequência	**Qtd. de *frames***	**Densidade de Pedestres**	**Ambiente**
PETS2009-S1L1-1-(13-57)	221	Médio	Nublado Uniforme
PETS2009-S1L1-2-(13-59)	241	Médio	Nublado Uniforme
PETS2009-S1L2-1-(14-06)	201	Alto	Nublado Uniforme
PETS2009-S1L2-2-(14-31)	131	Alto	Sombra e Sol
PETS2009-S2L1-(12-34)	795	Médio	Ensolarado Uniforme
PETS2009-S2L2-(14-55)	436	Médio	Sombra e Sol
PETS2009-S2L3-(14-41)	240	Alto	Sombra e Sol
PETS2009-S3MF1-(12-43)	107	Médio	Sombra e Sol

contendo pedestres caminhando em diferentes escalas devido às suas distâncias em relação à câmera, e com *backgrounds* desafiadores variados. A TUD-Campus e TUD-Crossing, em compressão png, foram preparadas em (ANDRILUKA; ROTH; SCHIELE, 2008), sendo que a TUD-Campus consiste de 71 imagens contendo 303 pedestres anotados e a TUD-Crossing composta de 201 imagens com 1008 pedestres. A TUD-Stadmitte, possui características mais similares à TUD-Campus, com 179 *frames* e foi apresentada em (ANDRILUKA; ROTH; SCHIELE, 2010). A Figura 35 ilustra *frames* dessas bases de dados utilizadas neste trabalho.

Tabela 3 – Principais características da Base de Dados TUD.

sequência	**Qtd. de *frames***	**Densidade de Pedestres**	**Ambiente**
TUD-Campus	71	Médio	Iluminação Uniforme
TUD-Crossing	201	Médio	Iluminação Uniforme
TUD-Stadmitte	179	Médio	Iluminação Uniforme

5.2 Indicadores de Avaliação

O objetivo de se utilizar indicadores de avaliação em trabalhos relacionados à detecção de pessoas é quantificar quão próximos às anotações de referências g_j^k, conhecidas como *ground truth*, está a resposta do detector d_j^k.

Na tentativa de avaliar os resultados de detecção de pessoas desta pesquisa, foram empregados os indicadores MODP (Precisão da Detecção de Múltiplos Objetos, do inglês, *Multiple Object Detection Precision*) e MODA (Assertividade da Detecção de Múltiplos Objetos, do inglês, *Multiple Object Detection Accuracy*), disponíveis em (STIEFELHAGEN

et al., 2007). Já para o fim de avaliar a precisão do extrator de características ASIFT foi empregado a métrica NMAE (Erro Absoluto Médio Normalizado, do inglês, *Normalized Mean Absolute Error*) (SHANI; GUNAWARDANA, 2011). Na prática, quando avaliados,

(a) PETS2009-S1L1-1-(13-57)

(b) PETS2009-S1L1-2-(13-59)

(c) PETS2009-S1L2-1-(14-06)

(d) PETS2009-S1L2-2-(14-31)

(e) PETS2009-S2L1-(12-34)

(f) PETS2009-S2L2-(14-55)

(g) PETS2009-S2L3-(14-41)

(h) PETS2009-S3MF1-(12-43)

Figura 34 – Imagens retiradas do Banco PETS2009.

(a) TUD-Campus

(b) TUD-Crossing

(c) TUD-Stadmitte

Figura 35 – Imagens retiradas dos Bancos TUD-Campus, TUD-Crossing e TUD-Stadmitte.

quanto maior forem os valores de MODP e MODA, máximo 1 (um), e menor forem os valores de NMAE, mínimo 0 (zero), melhor.

5.2.1 Anotações

Para um dado conjunto de *frames* I, a notação g_j^k representa a j-ésima anotação do *ground truth* de um objeto no k-ésimo *frame*. Em se tratando de anotação de pessoas, as informações existentes representam as características das *boxes*, ou caixas, que virtualmente representam o perímetro que circunda o posicionamento de uma pessoa na imagem. A Figura 36 mostra o conjunto *ground truth* de um *frame* que contém pedestres caminhando.

Sendo assim, o conteúdo de cada anotação corresponde às dimensões das *boxes*, que no caso do *ground truth* é dado por $g_j^k = \left\{x_j^k, y_j^k, w_j^k, h_j^k\right\}$. O par x_j^k e y_j^k correspondem às localizações do canto esquerdo superior das *boxes*, enquanto w_j^k e h_j^k representam a quantidade de *pixels* correspondentes à largura e altura das *boxes*, respectivamente.

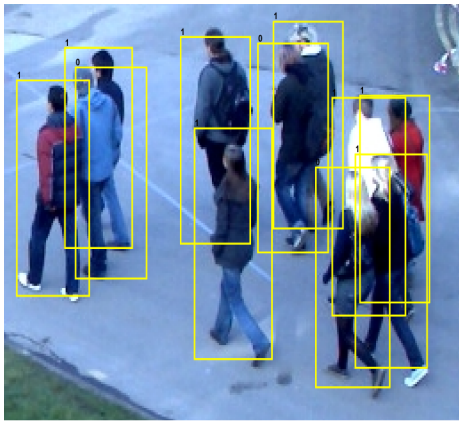


Figura 36 – Anotações *Ground truth* de um Conjunto de Pessoas.

Ainda existem as anotações d_j^k que revelam as *boxes* de saída do detector. O conteúdo deste padrão de anotação é o mesmo daquela mencionada para as *boxes ground truth*, com a diferença de que adiciona-se um termo notado como $score_j^k$, o qual refere-se à um valor numérico que expressa a precisão da k-ésima detecção do j-ésimo *frame* segundo a avaliação do algoritmo detector, ficando assim a notação $d_j^k = \left\{x_j^k, y_j^k, w_j^k, h_j^k, score_j^k\right\}$. Em linhas gerais, quanto maior o valor de $score_j^k$ maior a confiança na existência daquela detecção.

5.2.2 MODP

O indicador MODP, para a resposta de um detector, mostra a capacidade deste detector em estimar precisamente a posição dos objetos ou, no caso deste estudo, a posição das pessoas em uma sequência de *frames*.

Para o cálculo da MODP usa-se basicamente as informações de sobreposição espacial entre as informações de *ground truth* (g_j^k) e saída do sistema de detecção (d_j^k), como definido na Equação 5.1.

$$Overlap = \sum_{j=1}^{N^k} \frac{\left|g_j^k \bigcap d_j^k\right|}{\left|g_j^k \bigcup d_j^k\right|}, \tag{5.1}$$

onde N^k é o número de pessoas existentes no *frame* I^k.

Para cada *frame* I^k calcula-se o valor de $MODP^k$ pela Equação 5.2, que representa a precisão da localização das pessoas neste *frame*. Se $N^k = 0$, $MODP^k$ deve ser forçado para zero.

$$MODP^k = \frac{Overlap}{N^k}, \tag{5.2}$$

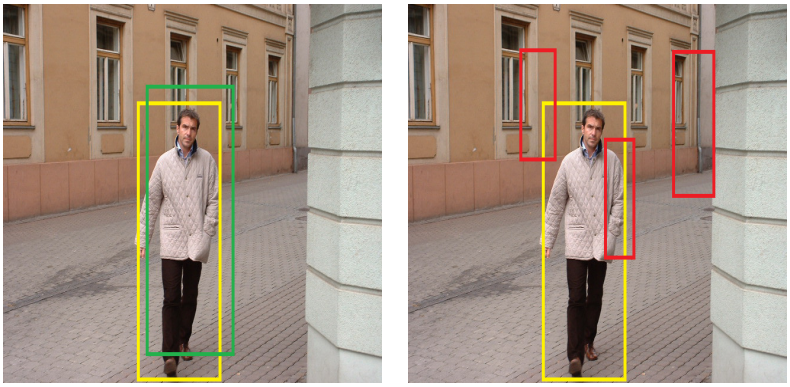
Já para uma sequência de *frames*, o valor de MODP é encontrado pela Equação 5.3.

$$MODP_{sequência} = \sum_{k=1}^{N_{frames}} \frac{MODP^k}{N_{frames}}, \tag{5.3}$$

onde N_{frames} representa o número de *frames* existentes na sequência que deseja-se calcular a MODP.

5.2.2.1 Perdas e Falsos Positivos em Detecções

No que se refere à detecções de objetos ou pedestres, as designações de perdas e falsos positivos são usadas para indicar o *status* de cada entidade de saída d do detector em relação ao *ground truth* g.



(a) Em amarelo, anotação *ground truth*. Em verde, detecção válida.

(b) Em amarelo, anotação *ground truth* reconhecida como uma perda. Em vermelho falsos positivos.

Figura 37 – Exemplo de avaliação em detecções de pessoas.

1. Perdas (p): Para um dado *frame* I^k, o valor de perdas (p^k) é a quantidade de anotações em g^k que não possuem correspondências em d^k. Neste trabalho, adota-se a padronização de (STIEFELHAGEN et al., 2007), onde as perdas são *boxes* em g^k que não apresentam sobreposição (*overlap*) de no mínimo 0,2 com alguma detecção em d^k.

2. Falsos Positivos (fp): Para um dado *frame* I^k, o valor de falsos positivos (fp^k), é a quantidade de detecções em d^k que não possuem correspondências em g^k. Um falso positivo é reconhecido como uma detecção em d^k que não tiver alguma sobreposição (*overlap*) de pelo menos 0,2 com alguma anotação em g^k.

Importante ressaltar que na avaliação das Perdas e Falsos Positivos busca-se o maior valor de sobreposições exclusivas entre os conjuntos g^k e d^k, ou seja, existe apenas uma correspondência válida no conjunto g^k em d^k e vice e versa.

A Figura 37 mostra exemplo de detecção válida, perda e falsos positivos.

5.2.3 MODA

O indicador de assertividade de detecções MODA, para a resposta de um detector, mostra a capacidade deste detector em estimar precisamente a quantidade de objetos ou, no caso deste estudo, a quantidade de pessoas em uma sequência de *frames*. Neste sentido, apenas valores de perdas (p) e falsos positivos (fp) são utilizados para os cálculos. Para um dado *frame* I^k, o valor da MODA é encontrado pela Equação 5.4

$$MODA(k) = 1 - \frac{p^k + fp^k}{N_G^k}, \tag{5.4}$$

onde N_G^k é o número de anotações *ground truth* do k-ésimo *frame*.

Para uma sequência de *frames*, a MODA normalizada é calculada pela Equação 5.5, relacionando todas as quantidades de perdas e falsos positivos da sequência.

$$MODA_{sequência} = 1 - \frac{\sum_{i=1}^{N_{frames}}(p_i + fp_i)}{\sum_{i=1}^{N_{frames}}(N_G^i)}. \tag{5.5}$$

5.2.4 NMAE

Existe uma preocupação especial em entender a precisão do extrator de características a ser utilizado. Quanto mais preciso for o extrator de características, melhor representado estará o movimento das pessoas pelo mapa de densidade encontrado pela abordagem descrita no Capítulo 3. Partindo disso, utiliza-se neste trabalho o indicador NMAE sobre a sequência de *frames* completa, o qual permite estimar quão bem o extrator de características define os *pixels* relevantes à pessoas disponíveis nas anotações g^k.

O cálculo do NMAE parte do conceito utilizado pelo MAE (Erro Absoluto Médio, do inglês, *Mean Absolut Error*), o qual permite avaliar a precisão de sistemas, como sistemas de predição por CF (Filtragem Colaborativa, do inglês, *Collaborative Filtering*) (GOLDBERG et al., 2001), que buscam prever o desejo de conteúdo de usuários na *internet*.

A cada novo *frame* I^k, disponibilizado para detecção de pedestres, novos mapas de densidade C^k e P^k são calculados para os *pixels* de características extraídas da imagem (Seção 3.1), e para os *pixels* centrais de cada *box* indicados pelas anotações em g^k, respectivamente. Para cada mapa de densidade então, encontra-se o valor de MAE pela Equação 5.6 a partir dos valores de $\widetilde{C}^k$, o qual é gerado pela regressão linear de C^k em relação a P^k.

$$MAE^k = \frac{1}{N_{pixels}} \sum_{j=1}^{N_{pixels}} \left| \widetilde{c}_j^k - p_j^k \right|, \tag{5.6}$$

onde N_{pixels} é o número de *pixels* do *frame* I^k; p_j^k são os valores de densidade para cada localização de *pixel* de P^k, e $\widetilde{c}_j^k$ são os valores de densidade estimados pela regressão linear de localização de cada *pixel* de $\widetilde{C}^k$.

A Figura 38 ilustra o fluxo da metodologia de avaliação dos mapas de densidade, sendo que o mapa de densidade *ground truth* (P^k) é calculado a partir das anotações das pessoas. Os valores de p_j^k do mapa de densidades *ground truths* (P^k) são plotados versus os valores $\widetilde{c}_j^k$ do mapa de densidade estimado ($\widetilde{C}^k$) visando encontrar a equação linear que mapeia os valores estimados para os valores de *ground truth*.

Por fim, após a disponibilidade do conjunto MAE^k, calculado para todos os *frames* da sequência do vídeo, é possível normalizar estes valores a fim de encontrar o valor

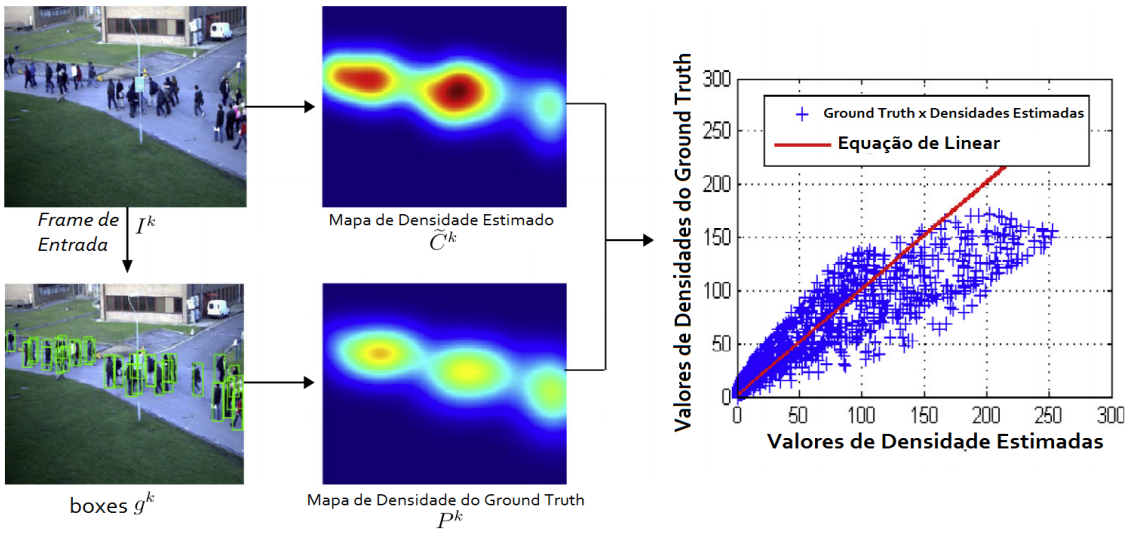


Figura 38 – Fluxo da metodologia de avaliação dos mapas de densidade de pessoas (Adaptado de (FRADI; DUGELAY, 2015)).

de NMAE. Essa normalização ocorre pela divisão da soma dos valores de MAE^k pela diferença dos valores máximos ($\widetilde{c}_{max}$) e mínimos ($\widetilde{c}_{min}$) de $\widetilde{c}_j^k$, como pode ser visto na Equação 5.7

$$NMAE = \frac{\sum_{k=1}^{N_{frames}} MAE^k}{\widetilde{c}_{max} - \widetilde{c}_{min}}, \tag{5.7}$$

5.3 Parâmetros de Configuração Utilizados

Uma etapa fundamental para se obter resultados aceitáveis a partir das técnicas utilizadas é o uso de parâmetros que melhor se adaptem às peculiaridades de cada base de dados. Isso vale tanto para os parâmetros de inicialização dos detectores de pedestres MDPM e LDCF, quanto para os parâmetros das filtragens realizadas sobre as *boxes* de saída destes detectores. Essa correta parametrização é importante, porque tanto as bases de dados quanto a saída dos detectores variam em função de suas concepções. A saída dos detectores de pedestres, por exemplo, diferem sobre os tamanhos das *boxes*, precisão de detecção, "falsos positivos" e quantidade de sobreposições. Já as bases de dados variam em termos de quantidade e concentração de pessoas, luminosidade, velocidade de caminhar, *background* e oclusões de partes.

De forma a experimentar valores compatíveis para testes das técnicas, para cada detector de pedestre e cada base de dados foi recolhida uma amostra de até 5% dos *frames* de cada base dados que pudessem representar de forma satisfatória aquela coleção de imagens. Essa coleção de *frames* selecionados foram testados com algumas variações de parâmetros, antes de finalmente rodar os algoritmos para todos os *frames* das bases de

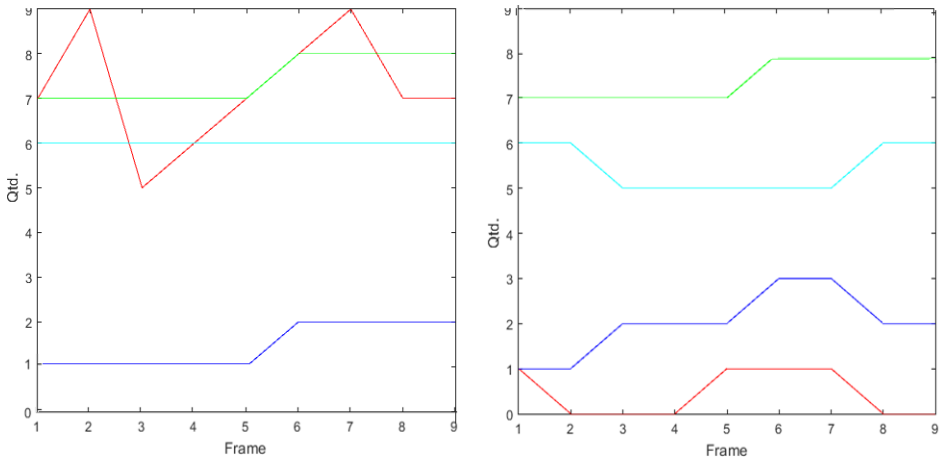
dados.

A Figura 39 ilustra o processo realizado para a base de dados TUD-Stadtmitte. Essa base de dados possui 179 *frames* e para isso foram usados 9 *frames* iniciais para ajuste de parâmetros que fornecessem respostas coerentes. Pela Figura 39a verifica-se a existência de um grande número de "falsos positivos" representados pela linha vermelha. Já na Figura 39b há uma forte redução de "falsos positivos" após a aplicação da técnica proposta com parâmetros específicos para essa base de dados. Mesmo havendo um aumento na quantidade de perdas (linha azul escuro) procura-se valores dos parâmetros que possam fornecer uma melhor relação de "perdas" e "falsos positivos" que colaborem com os resultados. A Figura 39 apresenta ainda informações de quantidade de "acertos" na cor ciano e a quantidade de anotações "*ground-truth*" na cor verde para os *frames*.

As variáveis parametrizáveis e seus valores utilizados neste trabalho são expostos na Tabela 4, para o detector MDPM, e na Tabela 5, para o detector LDCF.

Os parâmetros utilizados para os dois detectores (MDPM) e (LDCF) são diferentes, como visualizados nas Tabelas 4 e 5. A comparação entre os valores apresentados mostram o detector LDCF É mais robusto do ponto de vista que parâmetros únicos fornecem uma relação interessante nos resultados para todos os bancos de dados utilizados. No entanto, testes exaustivos sobre novos parâmetros, poderiam revelar alguma melhora específica para cada base de dados.

De forma consolidada, cada um dos parâmetros configuráveis utilizados nesta



(a) Saída Original do Detector MDPM sem integração com os mapas de densidade, e sem ajuste de parâmetros.

(b) Saída do Detector MDPM após integração dos mapas de densidade e com ajustes de parâmetros específicos

Figura 39 – Procedimento de Reconhecimento de Parâmetros. As linhas dos gráficos representam quantidades de : "falsos positivos" (vermelho), "acertos" (ciano), "*ground-truth*" (verde), "perdas" (azul).

Tabela 4 – Parâmetros utilizados para configuração dos testes com o detector MDPM.

	Detector MDMP		Filtros			Mapa de Densidade	*Forward-Backward*
	$\boldsymbol{t}$	$\boldsymbol{B}$	$\boldsymbol{\Delta_\gamma}$	$\boldsymbol{\Delta_0}$	$\boldsymbol{\Delta_h}$	$\boldsymbol{\sigma}$	$\boldsymbol{\zeta}$
PETS2009-S1L1-1-(13-57)	-1	0,5	0,02	0,5	100	90	1
PETS2009-S1L1-2-(13-59)	-1	0,5	0,02	0,5	100	90	1
PETS2009-S1L2-1-(14-06)	-1	0,5	0,02	0,5	100	90	1
PETS2009-S1L2-2-(14-31)	-1	0,5	0,02	0,5	100	90	1
PETS2009-S2L1-(12-34)	-1	0,5	0,003	0,4	50	90	1
PETS2009-S2L2-(14-55)	-1	0,5	0,02	0,5	100	90	1
PETS2009-S2L3-(14-41)	-1	0,5	0,02	0,5	100	90	1
PETS2009-S3MF1-(12-43)	-1	0,5	0,02	0,5	100	90	1
TUD-Campus	-1	0,5	0,003	0,4	50	40	1
TUD-Crossing	-1	0,5	0,003	0,4	50	40	1
TUD-Stadtmitte	-1	0,5	0,003	0,4	50	40	1

Tabela 5 – Parâmetros utilizados para configuração dos testes com o detector LDCF.

	Detector LDCF		Filtros			Mapa de Densidade	*Forward-Backward*
	$\boldsymbol{cascThr}$	$\boldsymbol{cascCal}$	$\boldsymbol{\Delta_\gamma}$	$\boldsymbol{\Delta_0}$	$\boldsymbol{\Delta_h}$	$\boldsymbol{\sigma}$	$\boldsymbol{\zeta}$
PETS2009-S1L1-1-(13-57)	-1	0,01	0,02	0,5	100	90	1
PETS2009-S1L1-2-(13-59)	-1	0,01	0,02	0,5	100	90	1
PETS2009-S1L2-1-(14-06)	-1	0,01	0,02	0,5	100	90	1
PETS2009-S1L2-2-(14-31)	-1	0,01	0,02	0,5	100	90	1
PETS2009-S2L1-(12-34)	-1	0,01	0,02	0,5	100	90	1
PETS2009-S2L2-(14-55)	-1	0,01	0,02	0,5	100	90	1
PETS2009-S2L3-(14-41)	-1	0,01	0,02	0,5	100	90	1
PETS2009-S3MF1-(12-43)	-1	0,01	0,02	0,5	100	90	1
TUD-Campus	-1	0,01	0,02	0,5	100	90	1
TUD-Crossing	-1	0,01	0,02	0,5	100	90	1
TUD-Stadtmitte	-1	0,01	0,02	0,5	100	90	1

pesquisa tem seu significado evidenciado abaixo:

- ***t***: *Detection Threshold*, ou Limiar de Detecção, em (FELZENSZWALB; MCALLESTER; RAMANAN, 2008), este parâmetro corresponde ao valor mínimo de *scores* no MDPM aceitáveis, ou seja, hipóteses com *score* menor do que t são descartadas.
- ***B***: *Overlap Threshold*, ou Limiar de Sobreposição, em (FELZENSZWALB; MCALLESTER; RAMANAN, 2008) corresponde ao valor máximo aceitável, no MDPM, de qualquer sobreposição entre *boxes* de hipóteses de pedestres de um mesmo *frame*.
- ***cascThr*** e ***cascCal***[5]: "constante de limiar de cascata" e "constante de calibração de cascata", respectivamente. Esses parâmetros afetam a precisão e a velocidade das iterações do detector LDCF. Ainda, é sugerido[5] que o valor de *cascThr* seja constante e igual a -1, e que seja variado o escalar *cascCal* até que a saída desejada seja encontrada.
- Δ_γ: Valor escalar de aceite para a restrição geométrica da Equação 4.1 apresentada na Subseção 4.1.1.
- Δ_h: Valor escalar de aceite de erro de altura de hipóteses preditas apresentado na Subseção 4.1.1.
- Δ_0: Valor limiar de sobreposição de *hipóteses*, apresentado na Subseção 4.1.2. Hipóteses com sobreposição maior do que Δ_0 são eliminadas.
- σ: Valor da "Largura de Banda" dos mapas de densidade utilizado na Equação 3.9 da Subseção 3.4.
- ζ: "Limiar de Distâncias". Características extraídas em um *frame* I^k, e que após a aplicação de fluxo óptico tiverem distâncias até sua localização correlacionada, no *frame* I^{k+1}, menores do que este limiar são eliminadas (Subseção 3.3).

5.4 Resultados da Aplicação da Metodologia

5.4.1 Precisão da ASIFT

O ganho esperado em relação ao uso da técnica de extração de características ASIFT está no seu potencial de viabilizar uma maior quantidade de características extraídas de imagens do que outras técnicas vistas na literatura. De forma visual, a Figura 23 apresenta um número muito maior de pontos detectados na utilização da ASIFT, sendo assim, espera-se uma maior precisão no parâmetro NMAE (Subseção 5.2.4) para está técnica.

[5] <https://github.com/pdollar/toolbox>

Tabela 6 – Valores de NMAE calculados para diferentes técnicas de extração de características na base PETS2009.

Base de Dados PETS2009	ASIFT	ASIFT sem Fluxo Óptico	FAST	SIFT	GFT	MSER	SURF
S1L1-1-(13-57)	**0,02**	0,09	0,07	0,07	0,08	0,07	0,07
S1L1-2-(13-59)	**0,03**	0,1	0,04	0,04	0,04	0,04	0,05
S1L2-2-(14-31)	**0,02**	0,04	0,09	0,09	0,10	0,07	0,07
S2L3-(14-41)	**0,01**	0,05	0,04	0,03	0,03	0,03	0,03

A Tabela 6 apresenta os resultados da métrica NMAE para a ASIFT e fluxo óptico calculados pelo autor em (LUCCHETTI; CIARELLI, 2016b) a partir de algumas bases de dados também utilizadas neste trabalho. Os resultados de NMAE encontrados foram comparados com outras técnicas de extração de características com e sem a filtragem por fluxo óptico descrito na Seção 3.3. O melhor resultado para cada vídeo está realçado em negrito. Como pode ser observado, o ASIFT foi superior em todos os casos.

Ainda pela Tabela 6, maiores valores de NMAE são verificados para os experimentos com ASIFT sem a utilização do processo de *forward-backward*, ou seja, uma menor precisão, caracterizando o importante papel do estágio *forward-backward* neste processo.

Neste trabalho também foram calculados NMAE para as demais bases de dados utilizadas, com e sem a mesma etapa de *forward-backward*. A Tabela 7 mostra a lista completa dos valores encontrados, sendo possível verificar que novamente os valores NMAE para a técnica completa com a filtragem por fluxo óptico é mais precisa. A comparação entre os valores da Tabela 7 e as características de densidade de pessoas das Tabelas 2 e 3 revela valores de NMAE menos precisos para bases de dados com um número médio de densidade de pessoas. Valores mais precisos de NMAE são visualizados para bases de dados com maior concentração de pessoas. Essa melhor filtragem está atrelada à maior quantidade de movimento concentrada nas regiões de alta densidade de pessoas, o que torna a técnica mais precisa para essas bases de dados pelos menores valores encontrados nas diferenças da Equação 5.6, que relaciona os *pixels* dos mapas de densidade estimados com os mapas de densidades calculados a partir dos centros das *boxes* anotadas no *ground-truth*.

Para exemplificar a resposta dos mapas de densidade gerados, as Figuras 40 e 41 apresentam mapas de densidade contruídos a partir dos *frames* das Figuras 34 e 35 que representam exemplos de imagens retiradas das bases de dados. As imagens representam de forma visual os movimentos encontrados pelos *keypoints* da ASIFT e filtrados pela abordagem *forward-backward* com o fluxo óptico.

Tabela 7 – Valores de NMAE calculados nas bases de dados utilizadas neste trabalho.

Base de Dados	NMAE	NMAE sem a etapa *forward-backward*
PETS2009-S1L1-1-(13-57)	**0,02**	0,09
PETS2009-S1L1-2-(13-59)	**0,03**	0,1
PETS2009-S1L2-1-(14-06)	**0,02**	0,03
PETS2009-S1L2-2-(14-31)	**0,02**	0,04
PETS2009-S2L1-(12-34)	**0,04**	0,06
PETS2009-S2L2-(14-55)	**0,03**	0,07
PETS2009-S2L3-(14-41)	**0,01**	0,05
PETS2009-S3MF1-(12-43)	**0,03**	0,1
TUD-Campus	**0,04**	0,09
TUD-Crossing	**0,06**	0,09
TUD-Stadtmitte	**0,07**	0,1

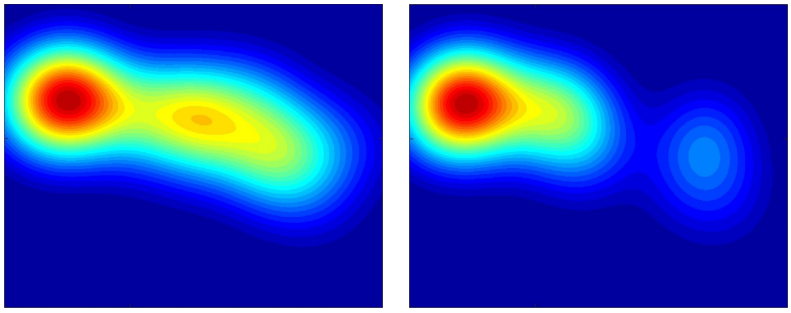
5.4.2 Resultados do Mapas de Densidade Sobre Detectores Individuais

A continuação da proposta deste trabalho compreende a interação de filtros e mapas de densidade, construídos a partir da ASIFT, com a saída de técnicas de detecção de pedestres.

Todos os códigos dos experimentos foram desenvolvidos em Matlab R2015a, em computador 64 bits com processador Intel *Core2Duo* 2,26 GHz e 4GB de RAM. Não houve em um primeiro momento foco na otimização de implementação das técnicas, sendo os esforços concentrados sobre os resultados das técnicas. Com o *hardware* mencionado, e com os parâmetros da Tabela 4 configurados, um ciclo completo de detecção de pedestres pela MDPM, para um *frame*, mais as filtragens por mapas de densidade fica perto de 7 segundos para o tempo de detecção, mais 4 segundos para a filtragem. Já para o LDCF, com os parâmetros da Tabela 5, o tempo de detecção de pedestres cai para 2 segundos, e 3 segundos para as filtragens por *frame*. Essa diferença entre o tempo de filtragem do MDPM para o LDCF se deve pela quantidade de hipóteses encontradas nas saídas destes detectores. O LDCF apresenta-se mais preciso, portanto, menos hipóteses a serem filtradas.

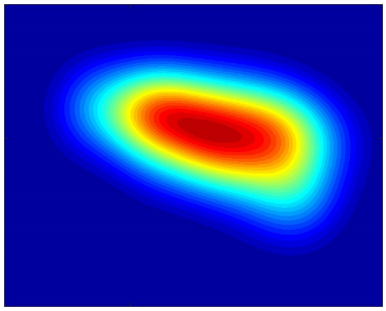
A interação entre a saída dos detectores de pedestres e os filtros e o mapa de densidade tem por expectativa a redução de hipóteses "falso positivas". Neste trabalho, as técnicas MDPM (Seção 2.3) e a LDCF (Seção 2.4) foram utilizadas para levantamento de hipóteses de pessoas nos vídeos. A interação entre mapas de densidade e a técnica MDPM já foi apresentada na literatura juntamente com seus limiares máximos e mínimos (Subseção 4.1.2) de *score* para cálculo dos mapas de densidade (FRADI; DUGELAY, 2015) e (LUCCHETTI; CIARELLI, 2016b). No entanto, para a LDCF tais valores de limiares

não estão disponíveis, assim, se fez necessário a geração de procedimento que pudesse suprir essa informação. A metodologia concebida para tal é aquela apresentada na Seção 4.2 pela

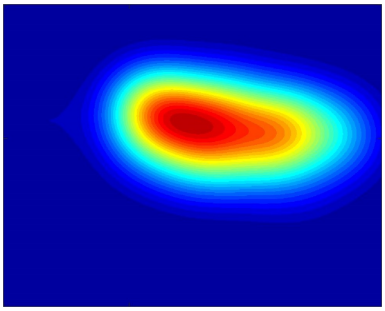


(a) PETS2009-S1L1-1-(13-57)

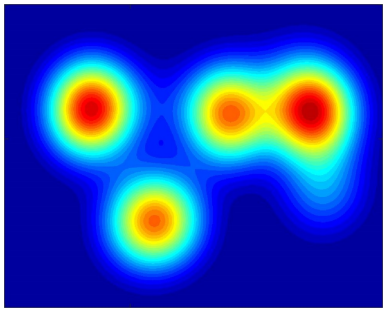
(b) PETS2009-S1L1-2-(13-59)



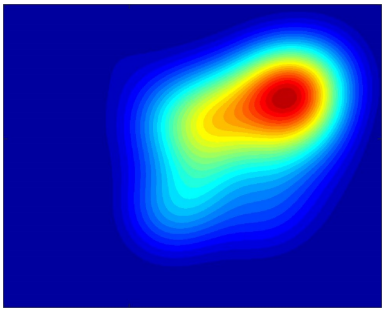
(c) PETS2009-S1L2-1-(14-06)



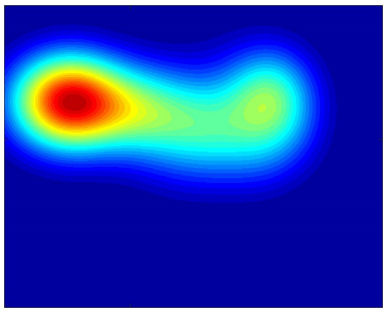
(d) PETS2009-S1L2-2-(14-31)



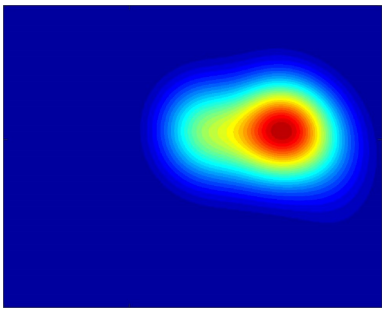
(e) PETS2009-S2L1-(12-34)



(f) PETS2009-S2L2-(14-55)

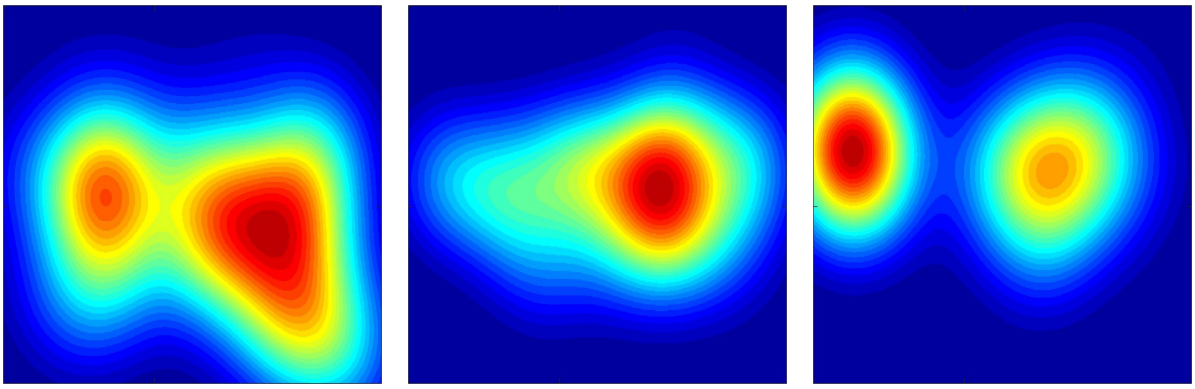


(g) PETS2009-S2L3-(14-41)



(h) PETS2009-S3MF1-(12-43)

Figura 40 – Mapas de densidade gerados a partir dos *frames* da Figura 34 e seus consecutivos *frames* "vizinhos".



(a) TUD-Campus (b) TUD-Crossing (c) TUD-Stadmitte

Figura 41 – Mapas de densidade gerados a partir dos *frames* da Figura 35 e seus consecutivos *frames* "vizinhos".

Tabela 8 – MODP e MODA obtidas da base de dados PETS2009 utilizando extratores de características distintos.

Dataset PETS2009	ASIFT with predicted detections		ASIFT without predicted detections		FAST		(without map)	
	MODP	MODA	MODP	MODA	MODP	MODA	MODP	MODA
S1L1-1-(13-57)	**0,65**	0,58	0,64	0,57	0,63	**0.63**	0,61	0,56
S1L1-2-(13-59)	**0,69**	**0,62**	0,68	0,60	0,68	0,6	0,66	0,56
S1L2-2-(14-31)	0,63	**0,49**	0,63	0,48	0,63	0,47	**0,69**	-17,53
S2L3-(14-41)	**0,66**	**0,49**	0,65	0,47	0,57	0,35	0,63	0,46

normalização dos *scores* e o cálculo de primeiro quartil dos *scores frame* a *frame*. Um passo adicional é ainda incluído visando uma possível recuperação de hipóteses que não foram identificadas pelos detectores (Seção 4.3). A Tabela 8, extraída de (LUCCHETTI; CIARELLI, 2016a), mostra ganhos encontrados por este último passo para algumas das bases de dados utilizadas também neste trabalho.

A Tabela 9 apresenta os resultados encontrados para o detector de pedestres MDPM, com os parâmetros da Tabela 4. A Tabela 9 contém os valores de MODP e MODA para todas as 11 bases de dados desta pesquisa incluindo as respostas para o valores de τ_{max} e τ_{min} fixos (τ "Não Adaptativo"), dado em (FRADI; DUGELAY, 2015), e adaptativos (τ "Adaptativo") pela técnica apresentada na Seção 4.2.

Um primeiro importante resultado visualizado na Tabela 9 são as melhoras visualizadas principalmente no indicado MODA, que relaciona diretamente a quantidade de "falsos positivos". Bases de dados como a PETS2009-S2L1-(12-34), PETS2009-S3MF1-(12-43) e as três bases TUD tiveram melhoras expressivas desse indicador atrelada à redução de "falsos positivos". O indicador de MODP também foi aprimorado em alguns casos, e isso indica uma melhor assertividade sobre as áreas anotadas de *ground-truth*. Ou seja,

foram removidas detecções pouco precisas do detector, mas que inicialmente haviam sido consideradas corretas pelo algoritmo de avaliação antes da aplicação das filtragens.

Um importante resultado extraído da Tabela 9 é a compatibilidade dos resultados de MODP e MODA entre a proposta do cálculo de limiares de τ_{max} e τ_{min} calculados dinamicamente *frame* a *frame* (Seção 4.2). Ainda, grande parte dos melhores resultados, evidenciados em negrito, são visualizados na coluna das respostas de limiares de τ encontrados de forma adaptativa.

A Tabela 10 apresenta os resultados encontrados para o detector de pedestres LDCF, com os parâmetros da Tabela 5. Nesta tabela não foram apresentados valores de "τ Não Adaptativo" por este valor não ter sido disponibilizado na literatura. Novamente, alguns ganhos são verificados quanto à MODA e à MODP, porém menos expressivos em magnitude do que àqueles encontrados na Tabela 9, isso devido à baixa quantidade de "falsos positivos" da própria técnica LDCF quando utilizada com os parâmetros da Tabela 5 sobre as bases de dados. Entretanto, verifica-se que boa parte dos melhores resultados de MODP, precisão do detector, estejam na coluna de "τ Adaptativo", ou seja, hipóteses imprecisas da resposta do detector foram removidas.

5.4.3 Resultados da Integração de Hipóteses de Detectores Filtrados

Uma etapa final proposta une as respostas filtradas dos detectores, e é avaliada sobre os ponto de vista dos mesmos indicadores aplicados às técnicas individuais (MODP e MODA). A Figura 42 ilustra de forma completa todo o processo de filtragem dos detectores até a integração das hipóteses.

Tabela 9 – Resultados de MODP e MODA para o Detector MDPM.

Base de Dados	Saída do MDPM		τ "Não Adaptativo"		τ "Adaptativo"	
	MODP	MODA	MODP	MODA	MODP	MODA
PETS2009-S1L1-1-(13-57)	0,610	0,561	**0,651**	0,580	0,629	**0,589**
PETS2009-S1L1-2-(13-59)	0,660	0,560	**0,690**	0,620	0,641	**0,643**
PETS2009-S1L2-1-(14-06)	0,586	0,471	0,586	**0,472**	**0,596**	0,449
PETS2009-S1L2-2-(14-31)	0,606	**0,460**	0,606	0,459	**0,618**	0,439
PETS2009-S2L1-(12-34)	**0,682**	0,084	0,670	0,582	0,677	**0,590**
PETS2009-S2L2-(14-55)	**0,651**	0,465	**0,651**	0,509	**0,651**	**0,513**
PETS2009-S2L3-(14-41)	0,660	0,460	0,660	0,490	**0,677**	**0,503**
PETS2009-S3MF1-(12-43)	**0,703**	-0,034	0,667	0,550	0,671	**0,605**
TUD-Campus	0,702	0,284	0,704	**0,696**	**0,707**	0,693
TUD-Crossing	0,709	0,198	**0,717**	0,697	**0,717**	**0,702**
TUD-Stadtmitte	0,722	0,065	0,753	**0,715**	**0,755**	0,712

Tabela 10 – Resultados de MODP e MODA para o Detector LDCF.

	Saída do LDCF		τ "Adaptativo"	
Base de Dados	MODP	MODA	MODP	MODA
PETS2009-S1L1-1-(13-57)	0,608	**0,521**	**0,620**	0,510
PETS2009-S1L1-2-(13-59)	0,599	**0,638**	**0,612**	0,620
PETS2009-S1L2-1-(14-06)	0,535	**0,456**	**0,539**	0,450
PETS2009-S1L2-2-(14-31)	0,619	**0,423**	**0,625**	0,410
PETS2009-S2L1-(12-34)	0,706	0,635	**0,722**	**0,710**
PETS2009-S2L2-(14-55)	0,682	**0,464**	**0,699**	0,430
PETS2009-S2L3-(14-41)	0,669	**0,460**	**0,680**	0,450
PETS2009-S3MF1-(12-43)	0,701	0,776	**0,711**	**0,870**
TUD-Campus	**0,730**	0,723	0,728	**0,750**
TUD-Crossing	0,786	0,795	**0,790**	**0,830**
TUD-Stadtmitte	0,825	0,710	**0,839**	**0,740**

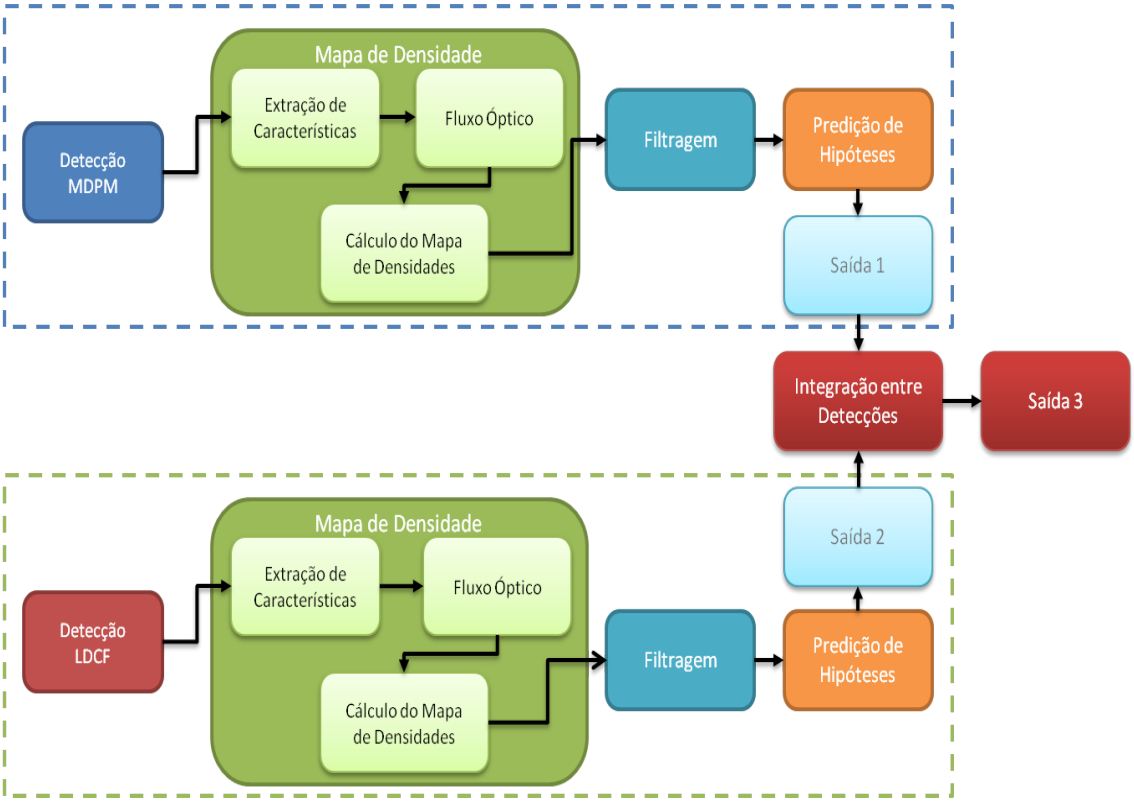


Figura 42 – Diagrama indicando a localização das modificações propostas em cada etapa do processo.

A Tabela 11 apresenta os resultados encontrados, até a etapa de integração, considerando os resultados da MDPM filtrados a partir de mapas de densidade com τ "Não Adaptativo", ou seja, $\tau_{max} = -0,5$ e $\tau_{min} = -1,2$, enquanto que a Tabela 12 mostra os

resultados para utilizações da MDPM filtrados a partir de mapas de densidade com τ "Adaptativo", proposto neste projeto. As duas Tabelas (11 e 12) foram construídas com o LDCF apresentando resultados a partir de τ "Adaptativo" e apresentam os resultados encontrados pela integração de cada processo.

A visão dada pelas Tabelas 11 e 12 são interessantes por evidenciar, em negrito, que os melhores resultados se encontram do lado direito das tabelas, ou seja, permeiam em grande parte sobre os resultados fornecidos pela integração das respostas das técnicas de detecção individuais. Mesmo que sejam adicionados "falsos positivos" a este processo de integração, há uma redução do número de "perdas", que balanceia os resultados de MODP e MODA. De qualquer forma, a mesma observação realizada pela Tabela 10 é válida aqui, onde os resultados de precisão encontrados pela LDCF são bastante competitivos e expressivos, principalmente sobre as bases de dados TUD, onde as hipóteses filtradas do detector LDCF apresentam baixos valores de "falsos positivos" pelos bons valores de MODA. No entanto, o mesmo não se concretiza totalmente para as bases de dados da PETS2009, que são mais desafiadoras.

Tabela 11 – Resultado com todos os dados até a Integração entre detectores com o MDPM utilizando mapas de Densidade com τ "Não Adaptativo".

	MDPM				LDCF				Resultado	
	Saída Original		τ "Não Adaptativo"		Saída do LDCF		τ "Adaptativo"		Detectores Integrados	
Base de Dados	MODP	MODA	MODP	MODA	MODP	MODA	MODP	MODA	MODP	MODA
PETS2009-S1L1-1-(13-57)	0,610	0,561	0,651	0,580	0,608	0,521	0,620	0,510	**0,653**	**0,649**
PETS2009-S1L1-2-(13-59)	0,660	0,560	**0,690**	0,620	0,599	0,638	0,612	0,620	0,652	**0,684**
PETS2009-S1L2-1-(14-06)	0,586	0,471	0,586	0,472	0,535	0,456	0,539	0,450	**0,589**	**0,558**
PETS2009-S1L2-2-(14-31)	0,606	0,460	0,606	0,459	0,619	0,423	0,625	0,410	**0,631**	**0,529**
PETS2009-S2L1-(12-34)	0,682	0,084	0,670	0,582	0,706	0,635	0,722	**0,710**	**0,726**	0,617
PETS2009-S2L2-(14-55)	0,651	0,465	0,651	0,509	0,682	0,464	**0,699**	0,430	0,685	**0,552**
PETS2009-S2L3-(14-41)	0,630	0,460	0,660	0,490	0,669	0,460	**0,680**	0,450	**0,680**	**0,605**
PETS2009-S3MF1-(12-43)	0,703	-0,034	0,667	0,550	0,701	0,776	**0,711**	**0,870**	0,704	0,589
TUD-Campus	0,702	0,284	0,704	0,696	0,730	0,723	0,728	**0,750**	**0,743**	0,587
TUD-Crossing	0,709	0,198	0,717	0,697	0,786	0,795	**0,790**	**0,830**	0,785	0,634
TUD-Stadtmitte	0,722	0,065	0,753	0,715	0,825	0,710	**0,839**	**0,740**	0,815	0,702

Por fim, a Tabela 13 ilustra de forma consolidada os mesmos resultados das integrações com a MDPM filtrada por τ "Adaptativo" e "Não Adaptativo" visualizados nas Tabelas 11 e 12. Os melhores resultados entre essas duas técnicas mostram que os resultados permeiam em maior parte na região das colunas que indica os resultados para os valores de τ "Adaptativo".

Tabela 12 – Resultado com todos os dados até a Integração entre detectores com o MDPM utilizando mapas de Densidade com τ "Adaptativo".

	MDPM				LDCF				Resultado	
	Saída Original		τ "Adaptativo"		Saída do LDCF		τ "Adaptativo"		Detectores Integrados	
Base de Dados	MODP	MODA	MODP	MODA	MODP	MODA	MODP	MODA	MODP	MODA
PETS2009-S1L1-1-(13-57)	0,610	0,561	**0,651**	0,580	0,608	0,521	0,620	0,510	0,644	**0,650**
PETS2009-S1L1-2-(13-59)	0,660	0,560	**0,690**	0,620	0,599	0,638	0,612	0,620	0,655	**0,680**
PETS2009-S1L2-1-(14-06)	0,586	0,471	0,586	0,472	0,535	0,456	0,539	0,450	**0,590**	**0,560**
PETS2009-S1L2-2-(14-31)	0,606	0,460	0,606	0,459	0,619	0,423	0,625	0,410	**0,638**	**0,530**
PETS2009-S2L1-(12-34)	0,682	0,084	0,670	0,582	0,706	0,635	0,722	**0,710**	**0,733**	0,682
PETS2009-S2L2-(14-55)	0,651	0,465	0,651	0,509	0,682	0,464	**0,699**	0,430	0,686	**0,561**
PETS2009-S2L3-(14-41)	0,630	0,460	0,660	0,490	0,669	0,460	0,680	0,450	**0,688**	**0,584**
PETS2009-S3MF1-(12-43)	0,703	-0,034	0,667	0,550	0,701	0,776	**0,711**	**0,870**	0,707	0,674
TUD-Campus	0,702	0,284	0,704	0,696	0,730	0,723	0,728	**0,750**	**0,738**	0,640
TUD-Crossing	0,709	0,198	0,717	0,697	0,786	0,795	**0,790**	**0,830**	0,772	0,691
TUD-Stadtmitte	0,722	0,065	0,753	0,715	0,825	0,710	**0,839**	**0,740**	0,816	0,730

Tabela 13 – Resultados de MODP e MODA após a Integração dos Detectores.

	Detectores Integrados MDPM c/ τ "Não Adaptativo"		Detectores Integrados MDPM c/ τ "Adaptativo"	
Base de Dados	MODP	MODA	MODP	MODA
PETS2009-S1L1-1-(13-57)	**0,653**	0,649	0,644	**0,650**
PETS2009-S1L1-2-(13-59)	0,652	**0,684**	**0,655**	0,680
PETS2009-S1L2-1-(14-06)	0,589	0,558	**0,590**	**0,560**
PETS2009-S1L2-2-(14-31)	0,631	0,529	**0,638**	**0,530**
PETS2009-S2L1-(12-34)	0,726	0,617	**0,733**	**0,682**
PETS2009-S2L2-(14-55)	0,685	0,552	**0,686**	**0,561**
PETS2009-S2L3-(14-41)	0,680	**0,605**	**0,688**	0,584
PETS2009-S3MF1-(12-43)	0,704	0,589	**0,707**	**0,674**
TUD-Campus	**0,743**	0,587	0,738	**0,640**
TUD-Crossing	**0,785**	0,634	0,772	**0,691**
TUD-Stadtmitte	0,815	0,702	**0,816**	**0,730**

6 Conclusão e Trabalhos Futuros

Este trabalho teve como objetivo investigar o uso de mapas de densidade construídos a partir da técnica ASIFT e fluxo óptico de forma a tornar mais precisa a identificação de pedestres em grupo. Recursos adicionais ainda foram propostos e mostraram bons resultados, como é o caso da integração da saída de detectores já filtrados pelos mapas de densidade.

As base de dados de vídeos utilizadas neste trabalho mostraram-se desafiadoras para os detectores utilizados e possuem capacidades específicas de informações, que variam em características como: quantidade de pedestres por *frame*, concentração de pessoas por área do *frame*, oclusões de pedestres, distância dos pedestres em relação à câmera, iluminação ambiente e *background*. Isso faz com que os detectores "percam" hipóteses ou agreguem "falsos positivos". Deve-se considerar também que as base de dados representam vídeos adquiridos a partir de câmeras estáticas, ou seja, com posicionamento fixo ao longo da sequencia de *frames*.

Os resultados obtidos mostraram que o extrator de características ASIFT consegue identificar mais pontos relevantes e o fluxo óptico ajuda a torná-lo mais preciso, sendo superior a outras abordagens comparadas, tornando o sistema mais preciso.

A etapa de normalização dos *scores* do detector MDPM mostrou resultados em um patamar bem próximo aos resultados mostrados quando se utilizam os limiares de *scores* fixos ("Não Adaptativos") propostos na literatura. Essa etapa de normalização abre margem para utilização de outros detectores integrados a mapas de densidade, o qual foi explorado neste trabalho.

Ainda sobre a etapa de integração entre a saída dos detectores, esta se mostrou robusta e com melhor desempenho em algumas das bases de dados. Essa integração tem por concepção uma redução das "perdas" de hipóteses de detecção, uma vez que garante que a resposta final, após a integração, contenha a união das hipóteses encontradas por ambos detectores, mas removendo detecções que representam um mesmo pedrestre pela técnica de clusterização hierárquica binária. Essa abordagem acaba sendo complementar aos mapas de densidade, que pela sua característica de filtragem, acaba eliminando "falsos positivos" mas também algumas hipóteses que corretamente indicam indivíduos. Uma observação relevante também refere-se à movimentação de pedestres em relação à câmera nas imagens pois, regiões das imagens com baixa movimentação de pessoas, por exemplo pedestre parado, gera uma região de baixa densidade no mapa de densidade; assim é possível que haja a remoção de um "verdadeiro positivo" quando realizada a integração entre detector de pedestres e os mapas de densidades.

O detector de pedestres por LDCF se mostrou competitivo em relação aos resultados demonstrados, sem mesmo a aplicação de filtros e integração com os resultados da MDPM. Essa afirmação fica mais forte para as bases de dados TUD, mas o mesmo não se concretiza totalmente para as bases de dados da PETS2009, que são mais desafiadoras. Neste caso, a técnica de integração proposta com τ "Adaptativo" se mostrou mais robusta pelos valores de MODP e MODA apresentados.

Uma vez que as técnicas utilizadas neste trabalho podem ser empregadas a qualquer outro detector, um desafio ainda previsto está relacionado à utilização de detectores ainda mais robustos para tornar a tarefa de identificação de pessoas em grandes concentrações mais precisa. A utilização de detectores com capacidade de reconhecimento de cabeças é uma opção interessante a se investigar. Quanto à predição de pedestres, poderia-se também explorar a utilização de outras técnicas de previsão como por exemplo Filtros de Kalman e Filtros de Partículas. A experimentação de novos limiares, tanto para os detectores, quanto para os filtros, pode melhorar ainda mais as precisões de MODA e MODP no que tange a remoção de "falsos positivos" e isso pode ser investigado em trabalhos futuros.

Referências

ALI, I.; DAILEY, M. N. Multiple human tracking in high-density crowds. *Image and Vision Computing*, v. 30, n. 12, p. 966 – 977, 2012. ISSN 0262-8856. Citado na página 18.

ANDRILUKA, M.; ROTH, S.; SCHIELE, B. People-tracking-by-detection and people-detection-by-tracking. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. [S.l.: s.n.], 2008. p. 1–8. ISSN 1063-6919. Citado na página 71.

ANDRILUKA, M.; ROTH, S.; SCHIELE, B. Monocular 3d pose estimation and tracking by detection. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2010. Citado na página 71.

BENENSON, R. et al. Seeking the strongest rigid detector. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Portland, Oregon: IEEE, 2013. Citado 2 vezes nas páginas 17 e 18.

BENENSON, R. et al. Pedestrian detection at 100 frames per second. In: *CVPR*. IEEE Computer Society, 2012. p. 2903–2910. ISBN 978-1-4673-1226-4. Disponível em: <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2012.html#BenensonMTG12>. Citado na página 18.

BENENSON, R. et al. Ten years of pedestrian detection, what have we learned? In: _____. *Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II*. Cham: Springer International Publishing, 2015. p. 613–627. ISBN 978-3-319-16181-5. Disponível em: <http://dx.doi.org/10.1007/978-3-319-16181-5_47>. Citado na página 22.

BOUGUET, J. yves. Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000. Citado na página 53.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995. ISSN 1573-0565. Disponível em: <http://dx.doi.org/10.1023/A:1022627411411>. Citado na página 26.

DALAL, N.; TRIGGS, B. Histograms of oriented gradients for human detection. In: *In CVPR*. [S.l.: s.n.], 2005. p. 886–893. Citado 8 vezes nas páginas 16, 17, 22, 24, 25, 33, 39 e 69.

DAVIES, A. C.; YIN, J. H.; VELASTIN, S. A. Crowd monitoring using image processing. *Electronics Communication Engineering Journal*, v. 7, n. 1, p. 37–47, Feb 1995. ISSN 0954-0695. Citado na página 18.

DAVIS, L. S. A survey of edge detection techniques. *Computer Graphics and Image Processing*, v. 4, n. 3, p. 248 – 270, 1975. ISSN 0146-664X. Disponível em: <http://www.sciencedirect.com/science/article/pii/0146664X7590012X>. Citado na página 45.

DOLLAR, P. et al. Fast feature pyramids for object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 36, n. 8, p. 1532–1545, ago.

2014. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2014.2300479>. Citado 6 vezes nas páginas 7, 18, 22, 41, 42 e 43.

DOLLAR, P. et al. Integral channel features. In: *Proceedings of the British Machine Vision Conference*. [S.l.]: BMVA Press, 2009. p. 91.1–91.11. ISBN 1-901725-39-1. Doi:10.5244/C.23.91. Citado 2 vezes nas páginas 17 e 18.

FELZENSZWALB, P.; MCALLESTER, D.; RAMANAN, D. A discriminatively trained, multiscale, deformable part model. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. [S.l.: s.n.], 2008. p. 1–8. ISSN 1063-6919. Citado 11 vezes nas páginas 7, 15, 17, 22, 33, 34, 35, 36, 37, 39 e 80.

FORSYTH, D. Object detection with discriminatively trained part-based models. *Computer*, v. 47, n. 2, p. 6–7, Feb 2014. ISSN 0018-9162. Citado na página 22.

FRADI, H.; DUGELAY, J.-L. Towards crowd density-aware video surveillance applications. *Information Fusion*, v. 24, p. 3 – 15, 2015. ISSN 1566-2535. Citado 13 vezes nas páginas 8, 15, 19, 44, 51, 56, 59, 61, 62, 63, 77, 82 e 84.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, v. 28, p. 2000, 1998. Citado na página 42.

FU, M. et al. Fast crowd density estimation with convolutional neural networks. *Engineering Applications of Artificial Intelligence*, v. 43, p. 81 – 88, 2015. ISSN 0952-1976. Citado 2 vezes nas páginas 16 e 19.

GARRIGUES, M.; MANZANERA, A. Real time semi-dense point tracking. In: _____. *Image Analysis and Recognition: 9th International Conference, ICIAR 2012, Aveiro, Portugal, June 25-27, 2012. Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 245–252. ISBN 978-3-642-31295-3. Disponível em: <http://dx.doi.org/10.1007/978-3-642-31295-3_29>. Citado na página 53.

GE, W.; COLLINS, R. T. Evaluation of sampling-based pedestrian detection for crowd counting. In: *Performance Evaluation of Tracking and Surveillance (PETS-Winter), 2009 Twelfth IEEE International Workshop on*. [S.l.: s.n.], 2009. p. 1–7. Citado na página 70.

GIBSON, J. J. *The perception of the visual world*. Boston: Houghton Mifflin, 1950. Disponível em: <http://opac.inria.fr/record=b1082917>. Citado na página 52.

GOLDBERG, K. et al. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, v. 4, n. 2, p. 133–151, 2001. ISSN 1573-7659. Citado na página 76.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. 1 p. ISBN 013168728X. Citado na página 14.

HARIHARAN, B.; MALIK, J.; RAMANAN, D. Discriminative decorrelation for clustering and classification. In: _____. *Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 459–472. ISBN 978-3-642-33765-9. Disponível em: <http://dx.doi.org/10.1007/978-3-642-33765-9_33>. Citado na página 41.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation*. 2nd. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998. ISBN 0132733501. Citado 3 vezes nas páginas 9, 26 e 32.

HEARST, M. A. Trends controversies: Support vector machines. *IEEE Intelligent System*, v. 13, n. 4, p. 18–28, 1998. Citado na página 31.

HORN, B. K.; SCHUNCK, B. G. Determining optical flow. *Artificial intelligence*, Elsevier, v. 17, n. 1-3, p. 185–203, 1981. Citado na página 53.

JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 22, n. 1, p. 4–37, jan. 2000. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/34.824819>. Citado na página 41.

JUNIOR, J. C. S. J.; MUSSE, S. R.; JUNG, C. R. Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine*, v. 27, n. 5, p. 66–77, Sept 2010. ISSN 1053-5888. Citado na página 14.

KHAN, F. S. et al. Color attributes for object detection. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*. [s.n.], 2012. p. 3306–3313. Disponível em: <http://dx.doi.org/10.1109/CVPR.2012.6248068>. Citado na página 17.

KHAN, R. et al. Discriminative color descriptors. In: *CVPR*. IEEE Computer Society, 2013. p. 2866–2873. ISBN 978-0-7695-4989-7. Disponível em: <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2013.html#KhanWKMDB13>. Citado na página 17.

LEVI, D.; SILBERSTEIN, S.; BAR-HILLEL, A. Fast multiple-part based object detection using kd-ferns. In: *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. [S.l.: s.n.], 2013. p. 947–954. ISSN 1063-6919. Citado na página 18.

LIM, J. J.; ZITNICK, C. L.; DOLLÁR, P. Sketch tokens: A learned mid-level representation for contour and object detection. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*. [s.n.], 2013. p. 3158–3165. Disponível em: <http://dx.doi.org/10.1109/CVPR.2013.406>. Citado na página 18.

LOWE, D. G. Object recognition from local scale-invariant features. In: *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*. [S.l.: s.n.], 1999. v. 2, p. 1150–1157 vol.2. Citado na página 45.

LUCAS, B. D.; KANADE, T. An iterative image registration technique with an application to stereo vision. In: *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981. (IJCAI'81), p. 674–679. Disponível em: <http://dl.acm.org/citation.cfm?id=1623264.1623280>. Citado na página 53.

LUCCHETTI, M.; CIARELLI, P. Combining density map and predict detections for pedestrian detection in crowds. In: *Workshop de Visão Computacional*. [S.l.: s.n.], 2016. Citado na página 84.

LUCCHETTI, M.; CIARELLI, P. Identificação de pedestres por meio de mapas de densidade construídos com asift e fluxo Óptico. In: *Congresso Brasileiro de Automática*. [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 81 e 82.

MA, W.; HUANG, L.; LIU, C. Crowd density analysis using co-occurrence texture features. In: *Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on*. [S.l.: s.n.], 2010. p. 170–175. Citado na página 18.

MARANA, A. N. et al. Estimation of crowd density using image processing. In: *Image Processing for Security Applications (Digest No.: 1997/074), IEE Colloquium on*. [S.l.: s.n.], 1997. p. 11/1–11/8. Citado na página 18.

MENZE, B. H. et al. On oblique random forests. In: *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 2011. (ECML PKDD'11), p. 453–469. ISBN 978-3-642-23782-9. Disponível em: <http://dl.acm.org/citation.cfm?id=2034117.2034147>. Citado na página 41.

MILITELLO, C.; RUNDO, L.; GILARDI, M. C. Applications of imaging processing to mrgfus treatment for fibroids: a review. *Translational Cancer Research*, v. 3, n. 5, 2014. ISSN 2219-6803. Disponível em: <http://tcr.amegroups.com/article/view/3200>. Citado 2 vezes nas páginas 8 e 54.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072. Citado na página 25.

MOISAN, L.; STIVAL, B. A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix. *International Journal of Computer Vision*, v. 57, n. 3, p. 201–218, 2004. ISSN 1573-1405. Disponível em: <http://dx.doi.org/10.1023/B:VISI.0000013094.38752.54>. Citado na página 51.

NAM, W.; DOLLAR, P.; HAN, J. H. Local decorrelation for improved pedestrian detection. In: GHAHRAMANI, Z. et al. (Ed.). *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014. p. 424–432. Disponível em: <http://papers.nips.cc/paper/5419-local-decorrelation-for-improved-pedestrian-detection.pdf>. Citado 6 vezes nas páginas 7, 15, 22, 41, 42 e 43.

NAM, W.; HAN, B.; HAN, J. H. Improving object localization using macrofeature layout selection. In: *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*. [s.n.], 2011. p. 1801–1808. Disponível em: <http://dx.doi.org/10.1109/ICCVW.2011.6130467>. Citado na página 17.

OTT, P.; EVERINGHAM, M. Implicit color segmentation features for pedestrian and object detection. In: *2009 IEEE 12th International Conference on Computer Vision*. [S.l.: s.n.], 2009. p. 723–730. ISSN 1550-5499. Citado na página 17.

PAISITKRIANGKRAI, S.; SHEN, C.; HENGEL, A. van den. Strengthening the effectiveness of pedestrian detection with spatially pooled features. *CoRR*, abs/1407.0786, 2014. Disponível em: <http://arxiv.org/abs/1407.0786>. Citado na página 18.

PARK, D.; RAMANAN, D.; FOWLKES, C. Multiresolution models for object detection. In: *Proceedings of the 11th European Conference on Computer Vision: Part IV*. Berlin, Heidelberg: Springer-Verlag, 2010. (ECCV'10), p. 241–254. ISBN 3-642-15560-X,

978-3-642-15560-4. Disponível em: <http://dl.acm.org/citation.cfm?id=1888089.1888108>. Citado na página 18.

RAO, A. S. et al. Estimation of crowd density by clustering motion cues. *The Visual Computer*, v. 31, n. 11, p. 1533–1552, 2014. ISSN 1432-2315. Citado na página 19.

RODRIGUEZ, J. J.; KUNCHEVA, L. I.; ALONSO, C. J. Rotation forest: A new classifier ensemble method. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 28, n. 10, p. 1619–1630, out. 2006. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2006.211>. Citado na página 41.

RODRIGUEZ, M. et al. Density-aware person detection and tracking in crowds. In: *Computer Vision (ICCV), 2011 IEEE International Conference on*. [S.l.: s.n.], 2011. p. 2423–2430. ISSN 1550-5499. Citado na página 16.

RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 2. ed. [S.l.]: Pearson Education, 2003. ISBN 0137903952. Citado 2 vezes nas páginas 26 e 27.

SACHTLER, W.; ZAIDI, Q. Visual processing of motion boundaries. *Vision Research*, v. 35, n. 6, p. 807 – 826, 1995. ISSN 0042-6989. Disponível em: <http://www.sciencedirect.com/science/article/pii/004269899400160N>. Citado na página 52.

SENST, T. et al. Cross based robust local optical flow. In: *21th IEEE International Conference on Image Processing*. Paris,France: [s.n.], 2014. p. 1967–1971. ISBN 978-1-4799-5750-7. Citado 3 vezes nas páginas 15, 53 e 56.

SENST, T.; EISELEIN, V.; SIKORA, T. Ii-lk – a real-time implementation for sparse optical flow. In: _____. *Image Analysis and Recognition: 7th International Conference, ICIAR 2010, Póvoa de Varzim, Portugal, June 21-23, 2010. Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 240–249. ISBN 978-3-642-13772-3. Disponível em: <http://dx.doi.org/10.1007/978-3-642-13772-3_25>. Citado na página 53.

SENST, T.; EISELEIN, V.; SIKORA, T. Robust local optical flow for feature tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE, v. 22, n. 9, p. 1377–1387, 2012. Citado na página 53.

SERMANET, P. et al. Pedestrian detection with unsupervised multi-stage feature learning. *CoRR*, abs/1212.0142, 2012. Disponível em: <http://arxiv.org/abs/1212.0142>. Citado 2 vezes nas páginas 18 e 22.

SHANI, G.; GUNAWARDANA, A. Recommender systems handbook. In: _____. Boston, MA: Springer US, 2011. cap. Evaluating Recommendation Systems, p. 257–297. ISBN 978-0-387-85820-3. Citado na página 72.

SINHA, S. N. et al. Gpu-based video feature tracking and matching. In: *EDGE, Workshop on Edge Computing Using New Commodity Architectures*. [S.l.: s.n.], 2006. v. 278, p. 4321. Citado na página 53.

SMOLA, A. J.; BARTLETT, P. J. (Ed.). *Advances in Large Margin Classifiers*. Cambridge, MA, USA: MIT Press, 2000. ISBN 0262194481. Citado na página 26.

STIEFELHAGEN, R. et al. Multimodal technologies for perception of humans, clear 2006, southampton, uk, april 6-7, 2006, revised selected papers. In: _____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. cap. The CLEAR 2006 Evaluation, p. 1–44. ISBN 978-3-540-69568-4. Citado 2 vezes nas páginas 72 e 75.

TUZEL, O.; PORIKLI, F.; MEER, P. Pedestrian detection via classification on riemannian manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 30, n. 10, p. 1713–1727, out. 2008. ISSN 0162-8828. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2008.75>. Citado na página 18.

VIOLA, P.; JONES, M. Robust real-time object detection. In: *International Journal of Computer Vision*. [S.l.: s.n.], 2001. Citado na página 18.

VIOLA, P.; JONES, M. J. Robust real-time face detection. *International Journal of Computer Vision*, v. 57, n. 2, p. 137–154, 2004. ISSN 1573-1405. Disponível em: <http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb>. Citado na página 22.

WANG, X.; HAN, T. X.; YAN, S. An hog-lbp human detector with partial occlusion handling. In: *ICCV*. IEEE Computer Society, 2009. p. 32–39. ISBN 978-1-4244-4419-9. Disponível em: <http://dblp.uni-trier.de/db/conf/iccv/iccv2009.html#WangHY09>. Citado na página 17.

WANG, X. et al. Regionlets for generic object detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 37, n. 10, p. 2071–2084, 2015. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2015.2389830>. Citado na página 17.

WOLF, L.; HASSNER, T.; TAIGMAN, Y. Similarity scores based on background samples. In: *Proceedings of the 9th Asian Conference on Computer Vision - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 2010. (ACCV'09), p. 88–97. ISBN 3-642-12303-1, 978-3-642-12303-0. Disponível em: <http://dx.doi.org/10.1007/978-3-642-12304-7_9>. Citado na página 17.

XIAOHUA, L.; LANSUN, S.; HUANQIN, L. Estimation of crowd density based on wavelet and support vector machine. *Transactions of the Institute of Measurement and Control*, v. 28, n. 3, p. 299–308, 2006. Disponível em: <http://tim.sagepub.com/content/28/3/299.abstract>. Citado na página 18.

YAN, J. et al. Robust multi-resolution pedestrian detection in traffic scenes. In: *CVPR*. IEEE Computer Society, 2013. p. 3033–3040. ISBN 978-0-7695-4989-7. Disponível em: <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2013.html#YanZLLL13>. Citado na página 18.

YANG, H. et al. The large-scale crowd analysis based on sparse spatial-temporal local binary pattern. *Multimedia Tools and Applications*, v. 73, n. 1, p. 41–60, 2012. ISSN 1573-7721. Citado na página 19.

YU, G.; MOREL, J.-M. Asift: An algorithm for fully affine invariant comparison. *Image Processing On Line*, v. 1, 2011. Citado 5 vezes nas páginas 8, 15, 48, 50 e 51.

YUILLE, A. L.; RANGARAJAN, A. The concave-convex procedure. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 15, n. 4, p. 915–936, abr. 2003. ISSN 0899-7667. Disponível em: <http://dx.doi.org/10.1162/08997660360581958>. Citado na página 33.

ZACH, C.; GALLUP, D.; FRAHM, J. M. Fast gain-adaptive KLT tracking on the GPU. In: *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition Workshops CVPRW '08*. [S.l.: s.n.], 2008. Citado na página 53.

ZENG, X.; OUYANG, W.; WANG, X. Multi-stage contextual deep learning for pedestrian detection. In: *2013 IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2013. p. 121–128. ISSN 1550-5499. Citado na página 18.

ZHANG, K.; LU, J.; LAFRUIT, G. Cross-based local stereo matching using orthogonal integral images. *IEEE Transactions on Circuits and Systems for Video Technology*, v. 19, n. 7, p. 1073–1079, July 2009. ISSN 1051-8215. Citado 4 vezes nas páginas 8, 53, 54 e 55.

ZHANG, S.; BAUCKHAGE, C.; CREMERS, A. B. Informed haar-like features improve pedestrian detection. In: *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. [S.l.: s.n.], 2014. p. 947–954. Citado na página 16.

ZHANG, Y. et al. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. *CoRR*, abs/1504.03293, 2015. Disponível em: <http://arxiv.org/abs/1504.03293>. Citado na página 17.

ZHOU, B.; ZHANG, F.; PENG, L. Higher-order {SVD} analysis for crowd density estimation. *Computer Vision and Image Understanding*, v. 116, n. 9, p. 1014 – 1021, 2012. ISSN 1077-3142. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1077314212000884>. Citado na página 18.